НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.855.5:519.257

До захисту допущено
В. о. завідувача кафедри ММСА

Оксана ТИМОЩУК

«___»_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра зі спеціальності 122 Комп'ютерні науки
на тему «Методи машинного навчання в сентимент аналізі текстової інформації»

Виконала:
студентка ІІ курсу, групи КА-83мн
Анна-Марія Павлівна _____

Науковий керівник:
Малишевський О. Г.,
старший викладач кафедри ММСА ІПСА
КПІ ім. Ігоря Сікорського, канд.техн.наук _____

Рецензент:

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань
Студент

_____

Київ
2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)
Спеціальність — 122 Комп’ютерні науки»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри ММСА
Оксана ТИМОЩУК
«___» _____2020 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Рудзевич Анні-Марії Павлівні

1. Тема дисертації: «Методи машинного навчання в сентимент аналізі текстової інформації», науковий керівник дисертації Малишевський Олексій Григорович, канд. техн. наук, затверджені наказом по університету від 07.04.2020 року № 959-с

2. Термін подання студентом дисертації: 13 травня 2020 р.

3. Об’єкт дослідження: сентимент аналіз текстової інформації

4. Предмет дослідження: алгоритми машинного навчання для задач визначення тональності тексту

5. Перелік завдань, які потрібно розробити:

- Дослідити існуючі підходи до аналізу тональності тексту;
- Провести огляд та обрати математичні методи для вирішення задачі сентимент аналізу;

– Здійснити моделювання;

– Проаналізувати ринкових можливостей запуску стартап-проекту;

– Зробити висновки за результатами наукового дослідження.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

– Приклади даних;

– Схеми побудови моделей;

– Таблиці та графіки результатів.

7. Орієнтовний перелік публікацій: опублікувати 1 наукову статтю у фаховому міжнародному науково-технічному журналі «Системні дослідження та інформаційні технології»

8. Дата видачі завдання: 10 березня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1	Оформлення концептуального вступу	10.03.2020—15.03.2020
2	Огляд технічної літератури за темою	16.03.2020—20.03.2020
3	Збір статистичних даних	21.03.2020—01.04.2020
4	Вибір методів аналізу	02.04.2020—07.04.2020
5	Розробка програмного комплексу	08.04.2020—25.04.2020
6	Аналіз отриманих результатів	26.05.2020—30.04.2020
7	Оформлення звіту, формулювання висновків	01.05.2020—05.05.2020

Студентка Анна-Марія РУДЗЕВИЧ

Науковий керівник дисертації Олексій МАЛИШЕВСЬКИЙ

РЕФЕРАТ

Магістерська дисертація: 88 с., 38 рис., 22 табл. і 39 джерела.

У магістерській дисертації досліджуються методи машинного навчання для задач сентимент аналізу.

Було розглянуто підходи до вирішення задачі сентимент аналізу і проведено огляд їх переваг та недоліків. Також описано основні методи МН для аналізу тональності тексту, а саме Наївний Байєсівський класифікатор, метод опорних векторів та згорткова нейронна мережа.

У роботі також розглянуто етапи попередньої обробки тексту, такі як стемінг, видалення стоп-слів, алгоритми переведення слів до векторної форми: мішок слів та TF-IDF векторайзер.

Практичне дослідження побудовано на аналізі коментарів з соціальної мережі Інстаграм для оцінки зміни громадської думки під час президентської передвиборчої кампанії 2019 року.

Наведені методи МН застосовуються для вирішення задачі сентимент аналізу з використанням актуальних попередньо оброблених даних. Отримані результати було проаналізовано та порівняно якість класифікації застосованих методів.

Ключові слова: машинне навчання, сентимент аналіз, аналіз тональності тексту, інтелектуальний аналіз тексту.

ABSTRACT

Master's Thesis: 88 pages, 38 figures, 22 tables and 39 sources

The methods of machine learning and its application to sentiment analysis are investigated in the master's thesis.

Approaches to solving the problem of sentiment analysis were considered and their advantages and disadvantages were reviewed. It also describes basic ML methods for analyzing text tonality, namely the Naive Bayes classifier, the support vector machine (SVM), and the convolutional neural network (CNN).

The thesis also deals with the stages of text pre-processing, such as stemming, stop-words removing and word embedding algorithms: bag-of-words, TF-IDF vectorizer.

The research is based on the analysis of Instagram comments to assess changes in public opinion during the 2019 Presidential election campaign.

The ML methods mentioned above are used for sentiment analysis of up-to-date pre-processed data. The obtained results were analyzed and the quality of the classification of the applied methods was compared.

Keywords: machine learning, sentiment analysis, opinion mining, text mining.

З М І С Т

ПЕРЕЛІК СКОРОЧЕНЬ

NLP – natural language processing

SVM – support vector machines

ЗНМ – згорткова нейронна мережа

МН – машинне навчання

НБК – наївний байєсівський класифікатор

ВСТУП

Із розвитком інформаційних технологій та стрімкого накопичення великих масивів даних широкого розповсюдження набула така область комп'ютерної лінгвістики, як сентимент аналіз. З розвитком даних технологій стало можливим автоматично витягати з тексту виражену автором думку, а також оцінювати текст як позитивний, негативний, а при необхідності виокремлювати конкретні емоції (радість, гнів, сум тощо). Для виокремлення емоційної оцінки автора застосовуються підходи з використанням тональних словників і правил або застосовують методи машинного навчання (МН).

Сентимент аналіз (аналіз тональності тексту) – це розділ глибинного аналізу даних (data mining) і область комп'ютерної лінгвістики, що займається вилученням думок та емоцій з текстових документів.

Хоча лінгвістика та обробка природних мов (NLP) мають давню історію, до 2000-х років майже не було досліджень, що стосуються сентимент аналізу. Але відтоді цю галузь вчені почали дуже активною вивчати. Цьому передує кілька причин, основна з них – величезна кількість даних у соціальних мережах та на просторах Інтернету. Крім цього, сентимент аналіз може бути широко застосований у комерційних проектах майже в кожній галузі. Це забезпечує сильну мотивацію для досліджень.

Термін «sentiment analysis» вперше був згаданий в роботі Nasukawa T. та Yi J. [1], а вираз «opinion mining» (аналіз думок) в роботі Dave K., Lawrence St. та Pennock D. [2]. Також значний вклад в розвиток сентимент аналізу внесли роботи Bo Pang and Lillian Lee [22] [23]. Загалом більшість досліджень стосувалися сентимент аналізу текстів англійською мовою. Для української мови такі дослідження почалися зовсім недавно.

Всі задачі пов'язані з обробкою природних мов є складними і неоднозначними. Загалом, задача визначення емоційної оцінки тексту є суб'єктивною, оскільки різні люди по-різному оцінюють одні й ті ж самі події і, відповідно, один і той самий текст. Також в тексті можуть бути присутні

орфографічні помилки, скорочення, абревіатури, сарказм, емоджі. Однакові слова вжиті в різному контексті можуть мати діаметрально протилежне емоційне навантаження. Все вищевказане перешкоджає створенню єдиної моделі, яка правильно класифікуватиме тональність тексту, незалежно від тематики.

У наш час сентимент аналіз набув широкого використання для маркетингових цілей, а саме для визначення думки клієнта про певний товар або послугу та краще орієнтувати своє повідомлення на цільову аудиторію. Також набуло популярності аналізування твітів, блогів, текстів новин, оглядів, коментарів для визначення відношення автора до суб'єкта його висловлення. Для цього застосовують різні методики, включаючи алгоритми обробки природних мов (NLP), статистику та методи машинного навчання.

У роботі ми застосовуємо сентимент аналіз для визначення настроїв користувачів стосовно кандидатів у Президенти України 2019. Ми аналізуватимемо коментарі користувачів в соціальній мережі Інстаграм протягом місяця до першого туру виборів на предмет позитивного або негативного ставлення до кандидата і зможемо оцінити як змінювалися настрої в суспільстві. Оскільки українці залишають коментарі як українською, так і російською мовами, ми будемо аналізувати ці дві мови.

Метою даної роботи є порівняння алгоритмів машинного навчання для аналізу тональності тексту на українській та російській мовах.

Для досягнення поставленої мети було поставлено ряд завдань, які визначають структуру дослідження, а саме:

- Дослідити існуючі підходи до аналізу тональності тексту;
- Провести огляд та обрати математичні методи для вирішення задачі сентимент аналізу;
- Реалізувати та порівняти алгоритми машинного навчання для оцінки тональності тексту;
- Проаналізувати ринкових можливостей запуску стартап-проекту;
- Зробити висновки за результатами наукового дослідження.

Дослідження аналізу настроїв мають важливий вплив не тільки на розвиток обробки природних мов, але також можуть мати глибокий вплив на науку управління, політологію, економіку та соціальні науки, оскільки вони сильно залежать від думки людей.

РОЗДІЛ 1. ОГДЯД ІСНУЮЧИХ ПІДХОДІВ ДО АНАЛІЗУ ТОНАЛЬНОСТІ ТЕКСТУ

Не зважаючи на актуальність даної задачі в наш час, ще на початку 2000-х років сентимент аналіз став популярним предметом для досліджень. За цей час було виокремлено рівні, на яких робиться аналіз, та розроблено декілька підходів до вирішення даної задачі. В цьому розділі ми розглянемо основні з них.

1.1. Рівні сентимент аналізу

Перш за все сентимент аналіз можна здійснювати на різних рівнях. Загалом виокремлюють три рівні [22]:

- рівень документу;
- рівень речення;
- рівень об’єкту та аспекту.

На рівні документу завданням сентимент аналізу є визначення позитивного чи негативного настрою документу загалом. Наприклад, система отримує відгук клієнта про певний товар і визначає, який сентимент висловлює цей текст про даний продукт. Цей рівень аналізу передбачає, що кожен документ висловлює думку лише про один суб’єкт (наприклад продукт) і не є придатним для документів, які оцінюють декілька суб’єктів.

Сентимент аналіз на рівні речення визначає, який настрій висловлює кожне окреме речення в документі: позитивний, нейтральний чи негативний. Нейтральний тон зазвичай означає, що жодної думки не було висловлено. Цей рівень аналізу тісно пов’язаний з класифікацією тексту на суб’єктивність [24], який визначає чи є висловлення об’єктивним (тобто таким, що висловлює фактичну інформацію), чи суб’єктивним (фраза висловлює думку, погляд).

Однак варто відзначити, що не лише суб’єктивні висловлення містять сентимент. Об’єктивні фрази також можуть містити приховану думку (наприклад: «Я купив новий смартфон і через два дні він перестав працювати»).

Аналіз на рівні документу та на рівні речення не дозволяє визначити, що саме подобається чи не подобається. На рівні об’єкту та аспекту у свою чергу можливо визначити сентимент, що стосується безпосередньо об’єкту висловлення. Цей рівень аналізу базується на ідеї, що сентимент (позитивний чи негативний) відноситься до певного об’єкту висловлення або до аспекту об’єкту. Одне речення може порівнювати два або більше об’єктів/аспектів, тому воно може містити декілька сентиментів. Наприклад речення «Мені все одно подобається цей фільм, хоча спецефекти погані». Загальний настрій позитивний, хоча все речення явно не є повністю позитивним. Якщо точніше, речення має позитивний сентимент стосовно фільму (об’єкт), але негативний стосовно його спецефектів (аспект фільму). Також можна виділити два типи думок: звичайні та порівняльні [25]. Звичайна думка висловлює настрій щодо одного об’єкту або його аспекту. Наприклад: «В Турції можна добре відпочити» висловлює позитивний сентимент про аспект Турції – відпочинок. А порівняльна думка порівнює декілька об’єктів за їх спільними аспектами. Наприклад в реченні: «В Італії відпочивати краще, ніж в Турції» порівнюються Італія та Турція за аспектом відпочинок і перевага надається Італії.

Отже, метою цього рівня аналізу є виокремлення сентименту від об’єкту висловлення або його аспекту.

Зважаючи на те, що класифікація сентименту як на рівні документу, так і на рівні речення вже є досить складним завданням, класифікація на рівні об’єкту ще більш проблематична. Окрім власне класифікації необхідно виокремити об’єкт(и) і/або аспект(и) сентименту, що не є простим завданням.

1.2. Підходи до вирішення задачі сентимент аналізу тексту

Оскільки семнтимент аналіз не є новим напрямком комп'ютерної лінгвістики, вчені напрацювали основні підходи до вирішення цієї задачі. Їх можна розподілити на [4]:

- підхід на основі правил;
- підхід на основі тональних словників;
- підхід з використанням методів машинного навчання (із вчителем і без нього).

1.2.1. Підхід, заснований на правилах

Зазвичай підходи, засновані на правилах, визначають набір правил у певній мові сценаріїв, що визначають суб'єктивність, полярність чи предмет думки.

Підхід, заснований на правилах, шукає думки в тексті і класифікує їх, базуючись на кількості негативних і позитивних слів. У ньому розглядаються різні правила класифікації, такі як слова-заперечення, що підсилюють значення слова, ідіоми, змішані думки і т.д. Самі правила будуються на основі поширених у мові шаблонів, закономірностей виділених з тексту. Це можуть бути слова і фрази.

Правило складається з антицедента та консеквента ({Антецедент} ⇒ {консеквент}), де антецедент описує умову і може бути одиночним шаблоном або серією шаблонів, з'єднаних оператором ∧, а консеквент позначає тональність, є результатом умови, що описує антецедент [7]. Цей метод має високу точність і використовується в комерційних системах, але складний в реалізації, тому що потрібна велика кількість правил для вірної класифікації:

необхідно врахувати всі можливі варіанти (варіантів стає в рази більше, якщо в мові порядок слів у реченні не фіксований).

Ця система дуже примітивна, оскільки не враховує, як слова поєднуються в послідовності. Можна зробити більш вдосконалену систему, але її складність почне швидко зростати. Крім того, додавання нових правил може мати небажані наслідки в результаті взаємодії з попередніми правилами. Також, ці системи потребують великих зусиль для підтримки існуючих і створення нових правил.

1.2.2. Підхід з використанням тональних словників

Для визначення тональності тексту також використовують тональні словники, які містять як загальні емоційно-забарвлені слова, так і вузько-спеціалізовану лексику. Кожному слову або словосполученню дається оцінка, що характеризує позитивний чи негативний сентимент. Часто такі словники складаються вручну або напів автоматизованими методами, але також є способи повністю автоматичного укладання тональних словників [38].

Виконуючи аналіз із застосуванням словників, кожному слову в тексті присвоюється певне значення тональності, взяте із тонального словника (якщо слово присутнє в словнику). Після цього визначається загальна тональність як середнє арифметичне всіх оцінок.

1.2.3. Підхід з використанням методів машинного навчання

Машинним навчанням є підрозділ штучного інтелекту, що займається алгоритмами, які дозволяють комп’ютерам навчатися. Алгоритму надається набір даних, з якого вилучається інформація про властивості даних. Ця інформація дозволяє йому робити прогнози на інших, раніше небачених даних.

Можливість робити прогнози щодо нових для моделі даних існує, оскільки майже всі невипадкові дані містять структури і шаблони, які дозволяють машинам робити узагальнення [20].

Машинне навчання також має слабкі сторони. Алгоритми відрізняються своєю здатністю узагальнювати великі набори шаблонів, і патерн, який не схожий на всі інші відомі алгоритму патерни, буде скоріше за все хибно класифіковано. Природна мова особлива тим, що в ній дуже багато шаблонів, які рідко зустрічаються. Тому, для досягнення точних результатів необхідно мати величезну навчальну вибірку, оскільки методи машинного навчання можуть лише обмежено робити узагальнення на основі даних, які вони вже бачили [20].

Машинне навчання зазвичай розрізняє три методи навчання: із вчителем, без вчителя та напівавтоматичне навчання (Semi-Supervised Learning). Навчання з підкріпленням (Reinforcement Learning) — це також метод машинного навчання, але він не використовується для класифікації тексту, тому ми не будемо розглядати його.

1.2.3.1. Машинне навчання із вчителем

Алгоритми машинного навчання класифікації тональності викликають інтерес через їх здатність моделювати багато признаків фіксуючи контекст, їх більш легку адаптацію до зміни вхідних даних та можливість вимірювати ступінь невизначеності. Найпоширенішим підходом при навчанні є використання одиничних слів (уніграм) у нижньому регістрі як признаків при описі навчальних та тестових прикладів.

Задача аналізу тональності тексту зазвичай моделюється як проблема класифікації, коли класифікатору подається на вхід текст у векторному вигляді і, який повертає відповідний клас, наприклад позитивний або негативний (у разі бінарної класифікації), позитивний, негативний чи нейтральний (у разі аналізу

полярності) або класи емоцій (радість, злість тощо) для ідентифікації конкретних емоцій.

Під час навчання модель вчиться асоціювати вхідні дані (текст) з відповідним висновком (тегом) на основі зразків з навчальної вибірки. Пари текстових векторів і тегів класів подаються на вхід алгоритму для навчання моделі.

Перший крок навчання класифікатора — перетворення тексту в числове значення, як правило, вектор. Зазвичай кожен компонент вектора представляє частоту слова або виразу у заздалегідь заданому словнику. Цей процес відомий як векторизація тексту, і класичним прикладом є «мішком слів» («Bag of Words"). Існує також його варіація "мішок n-грам", яка враховує порядок слів. Іншим алгоритмом векторизації слів є Word2Vec, який дає словам з подібним значенням подібне представлення у векторному просторі.

Підхід з використанням методів машинного навчання із вчителем дає відносно високу точність класифікації. Він полягає в тому, що на основі навчальної вибірки класифікатор самостійно виділяє признаки, що впливають на тональність. Таким чином, проблема залежності від предметної області вирішується шляхом використання навчальної вибірки з тієї ж області.

Простим рішенням є логістична регресія — вона швидко навчається навіть на великих наборах даних і забезпечує досить точні результати.

Інший хороший вибір моделі включає SVM, випадковий ліс (random forest) та наївний Байєсівський класифікатор. Ці моделі можна вдосконалити, якщо їх навчати не лише на окремих словах, а на біграмах чи триграмах. Це дозволяє класифікатору розрізняти заперечення та короткі фрази, які можуть містити сентимент, якого не мають окремо взяті слова. Звичайно, процес створення та тренування на n-грамах збільшує складність моделі і час навчання.

Поява глибокого навчання принесла багато загальних архітектурних моделей, які показують високу точності класифікації в задачах сентимент аналізу.

Згорткові нейронні мережі чудово підходять для виявлення сентименту. Ідея полягає в тому, що замість виконання згортків на пікселях зображень модель може замість цього виконувати ці згортки у векотризованому просторі ознак слів у реченні. Оскільки згортка відбувається послідовно, модель може розрізняти заперечення або n-грами, які несуть нову інформацію про тональність.

Рекурентні нейронні мережі (RNN) — одні з найбільш часто використовуваних моделей глибокого навчання для обробки природних мов. Оскільки ці мережі є зворотними, вони ідеально підходять для роботи з послідовними даними, такими як текст.

RNN можна також значно покращити за рахунок використання механізму уваги, який є окремо підготовленим компонентом моделі. Увага допомагає моделі визначити, на яких словах у послідовності тексту застосовувати її фокус, тим самим дозволяючи моделі консолідувати більше інформації за більше часових кроків.

1.2.3.2. Машинне навчання без вчителя

Навчання із вчителем зазвичай показує хороші результати, але інколи його застосування унеможливлює відсутність розмічених даних. Методи навчання без вчителя є ще одним варіантом машинного навчання, який не вимагає попередньо позначених даних.

Одним з важливих аспектів аналізу настроїв є те, що його можна виконувати, використовуючи моделі, що навчаються без вчителя, тобто без розмічених даних про тональність, лише текст. Ключовим моментом для навчання таких моделей з високою точністю є використання величезних масивів даних.

Методи навчання без вчителя автоматично знаходять закономірності в даних, не вимагаючи при цьому даних з мітками класів. Навчання без вчителя

також використовується для розмічення набору даних, який згодом може бути використаний для навчання із вчителем [21]. Прикладами методів навчання без вчителя є кластеризація — задача розділення об’єктів вибірки на декілька кластерів; та алгоритм максимізації очікування — алгоритм пошуку максимальної подібності екземплярів.

Кластеризація або кластерний аналіз відноситься до навчання без учителя, є задачею групування набору об’єктів таким чином, щоб об'єкти в одній групі (кластері) були певною мірою найбільш схожі один з одним, ніж об'єкти в інших групах (кластерах) [9]. Подібність об'єктів визначається за допомогою різних метрик. Перевагами цього підходу є цілковита автоматизація, відсутність навчальної вибірки. Але зазвичай такі моделі мають низьку точність, вимагають ручного коректування метрик подібності при поганих результатах. Оцінка точності кластеризації теж є нетривіальним завданням, оскільки є багато функціоналів оцінки якості кластеризації, але не існує загального і найбільш коректного [10].

Проміжну позицію між навчанням із вчителем та навчанням без вчителя займає напівавтоматичне навчання (Semi-Supervised Learning). Напівавтоматичне навчання — це підхід, який передбачає навчання класифікатора на невеликому розміченому наборі даних та на великому пулі немаркованих даних [19].

Який з описаних вище підходів застосовувати, залежить від кількох факторів: потрібної точності аналізу, наявності навчальних даних, часових рамок та цілей.

Головною проблемою всіх підходів, за винятком кластеризації, є прив’язка до мови: граматика правил, слова в шаблонах і словниках, навчальних вибірках. Також можливо поєднувати вищеперелічені підходи.

1.3. Особливості сентимент аналізу

1.3.1. Емотикони

Емотикон — це символ, що представляє настрій чи почуття. Емотикони відіграють важливу роль у визначенні сентименту висловлення. Вони несуть значне емоційне навантаження незалежно від тематики і мови. Оскільки емоції широко використовуються при написанні коментарів, буде недоречним виключати їх з речення для аналізу.

Тому перед початком аналізу необхідно провести попередню обробку тексту. Дослідники уклали емотикони у позитивні/негативні списки, що полегшує виявлення символів емоцій у наборі даних. Будь-яка емоція може бути позначена позитивною, якщо вона знайдена в позитивному списку E_p. Емоція позначається як негативна, якщо вона знайдена у списку негативних E_n. Позитивні та негативні списки можна записати як (1.1, 1.2):

$$E_p = \{\text{список позитивних емоцій}\} \quad (1.1)$$

$$E_n = \{\text{список негативних емоцій}\} \quad (1.2)$$

Оцінка сентименту та / або полярність обчислюється, як показано в (1.3)

$$Pol_{score}(e) = \begin{cases} -1 & \text{якщо } e \in E_n \\ 1 & \text{якщо } e \in E_p \\ 0 & \text{якщо } e \notin E_p \cap e \in E_n \\ 0 & \text{якщо } e \in E_p \cap e \in E_n \end{cases}, \quad (1.3)$$

Висновки до розділу

Аналіз настроїв та думок є однією з важливих дослідницьких тем комп'ютерної лінгвістики.

У першому розділі було визначено три рівні сентимент аналізу:

- рівень документу;
- рівень речення;
- рівень об'єкта та аспекту.

Також досліджено, які підходи застосовуються для задачі сентимент аналізу. Виділяють три загальноприйнятих підходи: на основі правил, на основі тональних словників та підхід з використанням машинного навчання (з учителем і без).

У першому підході аналіз тексту проводиться на основі завчасно складених правил. При виконанні сентимент аналізу із застосуванням словників, кожному слову в тексті присвоюється певне значення тональності, взяте із тонального словника (якщо слово присутнє в словнику). Потім обчислюється тональність, найчастіше як середнє арифметичне всіх значень.

В основі підходу з використанням МН лежать алгоритми, які вилучають патерни з даних і, на основі яких роблять прогнози на раніше невідомих даних. Такі алгоритми поділяються на ті, що навчаються з учителем і ті, що навчаються без нього. Перші дають досить хороші результати, але потребують розмічені дані для навчання. В свою чергу підхід з використанням алгоритмів МН без вчителя не потребує навчальної вибірки, але точність таких моделей нижче, ніж при навчанні з учителем.

РОЗДІЛ 2. МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ЗАДАЧІ СЕНТИМЕНТ АНАЛІЗУ

У цьому розділі ми розглянемо, які методи використовуються для попередньої обробки тексту та основні алгоритми для вирішення задачі аналізу тональності тексту.

2.1. Методи попередньої обробки тексту

Перш за все текст переводиться в масив слів і очищається від розділових знаків та стоп-слів. Для зменшення розмірності слова приводять до словотворчої форми, тобто відкидають закінчення, префікси та суфікси. Таким чином признаки узагальнюються і їх простіше класифікувати.

Для приведення слова до словотворчої форми використовують або лематизацію або стемінг. Лематизація враховує морфологію мови, тому точно визначає основу слова. Спочатку визначається частин мови, а потім застосовуються правила для відсікання закінчення і суфіксів в залежності від частини мови.

Стемінг в свою чергу не потребує словників, але і не гарантує співпадіння зі справжньою морфологічною основою слова. Цей алгоритм відрізає початок або кінець слова, спираючись на список префіксів та суфіксів.

2.1.1. Векторизація тексту

Щоб мати можливість працювати з текстом в рамках машинного навчання, його необхідно перевести до векторного вигляду. Розглянемо основні алгоритми для векторизації тексту.

2.1.1.1. Bag-of-Words

Нам потрібен спосіб представлення текстових даних для алгоритму машинного навчання, і модель «мішок слів» вирішує саме це завдання. Модель «мішок слів» (або Bag of Words) — це спосіб векторизації тексту для подальшого його використання в алгоритмах машинного навчання.

Мішок слів — це модель текстів на природній мові, в якій кожен документ або текст виглядає як невпорядкований набір слів без відомостей про зв’язки між ними. Його можна представити у вигляді матриці, кожен рядок в якій відповідає окремому документу, а кожен стовпець — окремому унікальному слову. На перетині рядка і стовпця міститься кількість входжень слова в відповідний документ.

У такому підході кожне слово або лексема називається «грама».

2.1.1.2. TF-IDF Vectorizer

Вага TF-IDF (Term Frequency-Inverse Document Frequency) — це статистичний показник, який використовується для оцінки того, наскільки важливим є слово для документа в колекції або корпусі. Важливість збільшується пропорційно кількості разів, коли слово з’являється в документі, але компенсується частотою слова в корпусі.

Частота Терміну (TF): це підрахунок частоти слова в поточному документі. Оскільки кожен документ відрізняється довжиною, можливо, що термін з’являтиметься частіше у довгих документах, ніж у коротких. Тому частота терміну ділиться на довжину документа для нормалізації

$$TF(t) = \frac{n_i}{\sum_k n_k}, \tag{2.1}$$

де n_i — кількість разів слово t зустрічається у документі, а в знаменнику кількість всіх слів у документі.

Зворотна частота документа (IDF): підрахунок того, наскільки рідко слово зустрічається в усіх документах. IDF — це міра того, наскільки рідкісним є цей термін. Чим більш рідкісний термін, тим більший показник IDF

$$IDF(t) = log\frac{|D|}{|d_i \supset t_i|}, \quad (2.2)$$

де $|D|$ — кількість документів в колекції, а $|(d_i \supset t_i)|$ — кількість документів, які включають слово t_i ($n_i \neq 0$).

2.1.1.3 Word2Vec

Word2Vec використовує просту нейронну мережу з одним прихованим шаром для навчання ваг. На відміну від більшості інших моделей машинного навчання, нас не цікавить прогноз цієї нейронної мережі, ключову роль відіграють саме ваги прихованого шару, які ми і будемо вчити. Вхідний вектор помножений на ці ваги і є векторним представленням слова.

Для створення представлень Word2Vec використовують два алгоритми – Skip-Gram та CBOW (Continuous Bag-of-Words) [16] [37]. Розглянемо ці алгоритми більш детально.

Skip-Gram

Модель Skip-Gram отримує на вхід слово і має передбачити ймовірність кожного слова в словнику бути сусіднім з вхідним словом. Тобто модель Skip-Gram передбачає контекст для слова, що було подане на вхід мережі.

Для того, щоб навчити нейронну мережу необхідно представити слова в числовій формі. Для цього використовують «one-hot-encoding» вектори, у яких

в позиції вхідного слова стоїть «1», а у всіх інших - «0». Таким чином, на вхід в нейронну мережу подається one-hot вектор, і на виході також отримуємо вектор розмірністю вхідного вектора, що містить для кожного слова словника ймовірність того, що це слово зустрінеться біля вхідного слова.

Дана нейронна мережа має один прихований шар. Вхідний вектор має розмірність 1xV, де V – кількість слів у словнику. Розмірність прихованого шару складає VxE, де E – гіперпараметр, який відповідає за розмір векторного представлення слова. Вихід з прихованого шару має розмірність 1xE, і підається у шар softmax. Розмір вихідного шару становить 1xV, де кожне значення у векторі буде оцінкою ймовірності цільового слова в цій позиції [17].

Маючи навчені ваги, ми отримуємо векторне представлення слова перемноживши вхідний вектор на ваги прихованого шару.

CBOW

Модель Continuous Bag-of-Words є протилежністю моделі Skip-Gram. Вона з огляду на контекст слів передбачає наскільки ймовірним для кожного слова у словнику, є те, що зустрінеться саме це слово.

Розміри прихованого і вихідного шару залишиться однаковим. Тільки розмірність вхідного шару та обчислення функцій активації прихованого шару зміниться. Якщо у нас є 4 контекстних слова для одного цільового слова, у нас буде 4 вхідних вектора 1xV. Кожен буде помножено на прихований шар VxE, повертаючи вектори 1xE. Всі чотири 1xE вектори будуть поелементно усереднені, щоб отримати остаточну активацію, яка потім подаватиметься у шар softmax [16].

2.2. Методи машинного навчання для вирішення задачі сентимент аналізу

Розглянемо більш детально алгоритми МН, які використовуються для сентимент аналізу тексту, а саме: Наївний Байєсівський класифікатор, Байєсові

мережі, метод максимальної ентропії, метод опорних векторів (SVM), згорткову нейронну мережу та дерева рішень.

2.2.1 Наївний класифікатор Байєса

Наївний Байєсівський класифікатор є ймовірнісним алгоритмом машинного навчання, заснований на теоремі Байєса, який широко використовується для задач класифікації.

Через відносну простоту реалізації НБК є одним з найпопулярніших алгоритмів для задач класифікації. Крім цього, він показує не гірші результати класифікації порівняно з більш складними алгоритмами.

Байєсівський класифікатор базується на формулі Байєса (2.3):

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{2.3}$$

Для задачі визнаечння тональності ми прогнозуємо ймовірність того, що документ d відноситься до класу c. В даному випадку, документ є вектором: $d = \{w_1, w_2, ..., w_n\}$, де w_i — вага i-ого терміна, а n — розмір словника. Тому згідно з теоремою Байеса маємо формулу (2.4):

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{2.4}$$

За формулою (2.4) обчислюється умовна ймовірність для всіх класів. Якщо умовна ймовірність приналежності документа d до класа c максимальна, то С є найбільш ймовірним класом, до якого належить документ d (2.5):

$$\mathrm{C} = argmax_c P(c|d), \tag{2.5}$$

Згідно з теоремою Байєса (2.6):

$$\mathrm{C} = argmax_c P(d|c) * P(c) \tag{2.6}$$

і оскільки $d = \{ w_1, w_2, \ldots, w_n \}$, то

$$\mathrm{C} = argmax_c P(w_1, w_2, \ldots, w_n|c) * P(c), \tag{2.7}$$

Знаменник може бути опущений, так як для одного і того ж документа d ймовірність $P(d)$ буде однаковою, отже, її можна не враховувати.

Наївний байєсівський класифікатор спирається на припущення, що всі ознаки $x_1, x_2, \ldots, x_n$ документа d не залежать один від одного. Класифікатор називається наївним саме через це припущення, яке насправді не відповідає дійсності. Але в будь-якому випадку результати класифікації досить високі.

Також припускається, що позиція слів у реченні не має значення. Тому, умовну ймовірність (2.8)

$$P\ (w_1, w_2\ , \ldots, w_n \mid c), \tag{2.8}$$

для признаків $x_1, x_2, \ldots, x_n$, можна представити як (2.9):

$$P\ (w_1 \mid c)\ *\ (w_2 \mid c)\ *\ldots*\ (\ w_n \mid c)\ =\ \Pi_i\ P(w_i \mid \mathrm{c}), \tag{2.9}$$

Таким чином, для знаходження найбільш ймовірного класу для документа $d = \{w_1, w_2, \ldots, w_n\}$ за допомогою наявного Байєсового класифікатора, необхідно порахувати умовні ймовірності приналежності документа d для кожного з представлених класів окремо і вибрати клас, який має максимальну ймовірність:

$$C_{NB} = argmax_c[P(c_j) * \prod_i P(w_i|c_j)] , \quad (2.10)$$

Далі оцінимо ймовірність класу $P(c_j)$. Вона є відношенням кількості докуметів класу *j* в навчальній вибірці до загальної кількості документів

$$P(c) = \frac{D_c}{D} \quad (2.11)$$

де D_c — кількість документів класу *c*, а *D* — загальна кількість документів у вибірці.

Щоб оцінити умовні ймовірності для ознак $\hat{P}(w_i \mid c_j)$, використовуватимемо таку формулу:

$$\hat{P}(w_i|c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)} \quad (2.12)$$

де $\hat{P}(w_i \mid c_j)$ є відношенням кількості слів w_i в класі c_j до загальної кількості слів у цьому класі, а *V* — кількість слів у словнику навчальної вибірки.

Однак, якщо в тестовому наборі зустрінеться слово, яке не зустрічалося в наборі навчальних документів, то ймовірність $P(w_i \mid c_j)$ цього слова для будь-якого з класів буде дорівнювати нулю. Оскільки $P(d \mid c_j) \approx \prod_i P(w_i \mid c_j)$, то і ймовірність приналежності документа до будь-якого з класів також буде дорівнює нулю, що є неправильним. Для вирішення цієї проблеми зазвичай використовують так зване аддитивное згладжування (add-1 smoothing або згладжування Лапласа). Ідея add-1 згладжування полягає в тому, що до частот появи всіх термінів зі словника штучно додається одиниця. Виходить, що терміни, які не були присутні в документах навчальної вибірки, отримують незначну, але не нульову ймовірність появи і, тим самим, дають можливість визначити документ в один із класів

$$\hat{P}(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in V}(count(w, c) + 1)} \quad (2.13)$$

де V — кількість слів у словнику навчальної вибірки.

Документ для класифікації подається у вигляді вектора: $d = \{ w_1, w_2, ..., w_n \}$, где $w_1, w_2, ..., w_n$ – ваги для кожного з термінів словника вибірки. w_i може бути кількістю входжень терміна x_i в документ d, або ж може бути задано бінарно. Для бінарного вектора число входжень терміна w_i не має значення, важливий лише факт появи w_i в документі d.

Можна помітити, що для відносно великих текстів, ймовірність $P(c_j \mid d)$ являє собою добуток великої кількості дуже маленьких дробів. Для того щоб уникнути втрати точності, можна замінити добуток ймовірностей сумою логарифмів ймовірностей [39].

2.2.2 Байєсова мережа

Основне припущення наївного Байєсівського класифікатора – це незалежність ознак. Іншою крайністю є припущення, що всі ознаки залежать одна від одної. Це припущення властиве Байєсовій мережі, яка представляє собою спрямований ациклічний граф, вузли якого є випадковими величинами, а ребра – умовними залежностями. Для задач аналізу тексту Байєсові мережі рідко використовуються через велику складність обчислень [10].

2.2.3 Метод максимальної ентропії

Метод максимальної ентропії (Maximum Entropy Classifier) перетворює мічені набори ознак у вектори за допомогою кодування. Цей кодований вектор потім використовується для обчислення ваг для кожної ознаки, які потім

можуть бути об'єднані для визначення найбільш ймовірної мітки для набору ознак. Цей класифікатор параметризований набором X{$weights$}, який використовується для об'єднання спільних ознак, що генеруються з набору ознак X{$encoding$}. Зокрема, кодування відображає кожну пару C{($featureset, label$)} у вектор. Потім ймовірність кожної мітки обчислюється за допомогою наступного рівняння:

$$P(fs|label) = \frac{dotprod(weights, encode(fs, label))}{\sum(dotprod(weights, encode(fs, l)\ forlinlabels))}$$

Кауфман [11] використовував класифікатор ME для виявлення паралельних речень між будь-якими мовними парами використовуючи невелику кількістю даних для навчання. Інші інструменти, розроблені для автоматичного вилучення паралельних даних з непаралельних корпусів даних, використовують специфічні для мови методики або потребують великих обсягів даних для навчання. Їх результати показали, що класифікатори ME можуть давати корисні результати майже для будь-якої мовної пари. Це може дозволити створення паралельних корпусів для багатьох нових мов.

2.2.4. Метод опорних векторів

Метод опорних векторів шукає гіперплощину, яка найкраще розділить дану вибірку на два класи. Допускається багатокласова класифікація стратегіями one-vs-all і one-vs-one.

Дано вибірку елементів $x_i \in \mathbb{R}n$ і зіставлені їм класи $y_i \in$ {-1,1}. Об'єкти вибірки представляються точками. Опорні вектори – це точки даних, розташовані ближче всього до гіперплощини, при їх видаленні зміниться положення гіперплощини [47]. Їх вважають критичними елементами набору даних. У простій задачі бінарної класифікації, з вибіркою, що лінійно розділяється, гіперплощину можна представити у вигляді лінії, що розділяє вибірку на два класи. Чим далі дані лежать від гіперплощини, тим коректніше

вони класифіковані. Кращою гиперплощиною вважається та, відстань 1/‖w‖ від якої до кожного класу є максимальною, де w – нормальний вектор до розділяючої гиперплощини, яка може бути записана як безліч точок x, що задовольняють рівняння (2.14)

$$wx - b = 0, \tag{2.14}$$

де b — допоміжний параметр.

Якщо навчальна вибірка лінійно роздільна, можна вибрати дві паралельні гіперплощини так, щоб вони розділили цю множину на два класи. Область між ними називається зазором, маржею. Ці площини описуються рівняннями (2.15)

$$\begin{aligned} wx - b &= 1 \\ wx - b &= -1, \end{aligned} \tag{2.15}$$

Мінімізуючи відстань $\|w\|$ і одночасно виключаючи потрапляння даних в зазор, отримуємо задачу мінімізації (2.16):

$$\|w\|^2 \rightarrow min$$

$$y_i\,(wx_i - b) \geq 1, \text{для } 1 \leq i \leq n \tag{2.16}$$

Таку задачу вважають еквівалентною пошуку сідлової точки функції Лангранжа, і зводять до задачі квадратичного програмування, де присутні лише двоїсті змінні λ_i.

Вирішивши дану задачу, можна висловити w і b формулами (2.17) і (2.18), відповідно:

$$w = \sum_{i=1}^{n} \lambda_i c_i x_i\,, \tag{2.17}$$

$$b = w \cdot x_i - c_i, \quad \lambda_i > 0, \tag{2.18}$$

Кінцевий класифікатор записується як (2.19)

$$a(x) = sign\left(\sum_{i=1}^{n} \lambda_i c_i x_i \cdot x - b\right), \quad (2.19)$$

Якщо вибірка лінійно нероздільна, відбувається відображення векторів у простір більшої розмірності. При цьому, в наведеній вище формулі (2.19) відбувається заміна скалярного добутку на одну з функцій нелінійного ядра $K(x_i, x)$. Після чого також будується найкраща розділяюча гіперплощина.

2.2.5. Згорткова нейронна мережа

Згорткова нейронна мережа (ЗНМ) найчастіше використовується для задач розпізнавання зображень. Але, незважаючи на це, вона показує гарні результати в задачах аналізу тексту. ЗНМ складається з шарів різних видів: згортковий, агрегувальний (пулінг) та повнозв'язний шари [19].

В операції згортки відбувається множення поелементно фрагмента шару, що представляє матрицю, на матрицю (ядро згортки або значення фільтра) ваг, результат підсумовується і записується в відповідну позицію наступного шару, формуючи сигнал активації для нейрона. Ядро згортки «рухають» з деяким кроком по всьому шару, що є на першому кроці вихідним зображенням. Самих ядер згортки декілька, кожне представляє кодування певної ознаки. Утворені в результаті шари демонструють наявність цієї ознаки в попередньому шарі і його координати, їх називають картами ознак.

До результату кожної операції згортки – числу, застосовується функція активації ReLU, яка прирівнює від'ємні скалярні величини до нуля (2.20):

$$ReLU\ (x) = max\ (0, x), \quad (2.20)$$

Пулінг шар має ту ж кількість карт, що і попередній згортковий шар, його мета — зменшити розмірність отриманих карт ознак. Фільтр (ядро) даного шару зазвичай має розмір 2×2, що дозволяє зменшити карти наступного шару в 2 рази, сканує карту поточного шару, вибираючи максимальне значення ячейки (Max Pooling).

Для однієї карти це продемонстровано на рис. 2.1.

Після проходження однакової кількості згортальних і субдискретизуючих шарів залишається великий набір карт, що зберігають абстрактні ознаки вихідного зображення. Вони передаються в повнозв'язну нейронну мережу, яка також може мати кілька шарів. Вихідний шар виводить N-мірний вектор, застосовуючи функцію активації, де N-число класів у завданні класифікації, координати — ймовірність приналежності до класу.

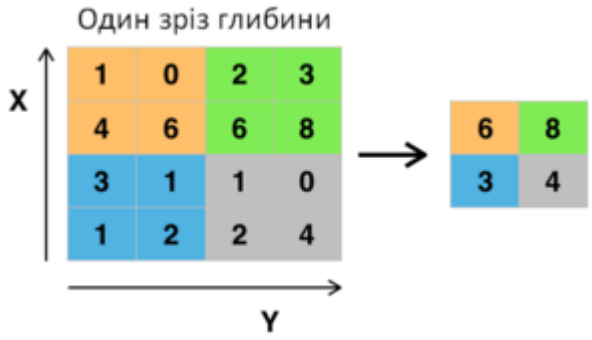

Рис. 2.1. Пулінг шар

Архітектура згорткової нейронної мережі для класифікації тексту бере за основу звичайну ЗНМ, але дещо спрощену. На вхід подається матриця, число її рядків дорівнює числу слів n в реченні (або документі), число стовпців – розмірності k векторного представлення слів. Для отримання нової ознаки виконується операція згоритки. Згортка полягає в застосуванні фільтра з вагами w на вікні з h слів. Ознака c_i генерується з вікна слів $x_{i:\,i+h-1}$ по формулі (2.21):

$$c_i = f\,(w * x_{i:\,i+h-1} + b), \tag{2.21}$$

де $b \in \mathbb{R}$ — нейрон зміщення, f — нелінійна функція, w — вектор ваг, $x_{i:\,i+h-1}$ – ковзне вікно.

Фільтр буде застосований до всіх можливих вікон слів у реченні $\{x_i{:}\ h, \ldots, x_{n-h+1} : n\}$ для отримання карти ознак:

$$c(w) = [c_1, c_2, \ldots, c_{n-h+1}] \tag{2.22}$$

Потім застосовується фільтр Max Pooling (максимізаційне агрегування), тобто шукається максимум у всій послідовності. Його ідея полягає в тому, щоб виділити найважливішу ознаку з самим високим значенням по кожній карті ознак:

$$\hat{c} = max(\mathrm{c}(w)) \tag{2.23}$$

Отримані таким чином значення передаються в повнозв’язний шар з функцією активації softmax, на виході отримуємо розподіл ймовірності по класах

$$P(y = j|x) = \frac{e^{x^T w_j + b_j}}{\sum_{k=1}^{K} e^{x^T w_j + b_j}} \tag{2.24}$$

Для запобігання перенавчанню на цьому шарі використовується метод виключення (дропаут) нейронів з ймовірністю p і $l2$-регулізація.

Зазвичай навчання мережі відбувається з використанням стохастичного градієнтного спуску. Використання дропауту вилучає з нейронної мережі деяку кількість нейронів (на етапі навчання) для запобігання коадаптації нейронів і, в результаті, отримання кращої узагальноючої здатності мережі. Вихід після використання дропауту можна представити у вигляді:

$$y = w(zr) + b, \quad (2.25)$$

де $z = [\hat{c}_1, \dots, \hat{c}_m]$, r — вектор, що містить 0 та 1.

Дропаут також прискорює процес навчання.

В якості гіперпараметрів мережі виділяють фільтр вікна (ядро) висоти h з деякою кількістю карт ознак, ймовірність дропаута p, $l2$-регулізацію і розмір батча. L2-регулязація штрафує ваги мережі, зменшуючи їх значення, і використовується для запобігання її перенавчання. Батч використовується для прискорення навчання, представляючи собою «пакет» випадково обраних ознак в методі стохастичного градієнтного спуску.

Загалом ЗНМ для аналізу тексту відрізняються від ЗНМ для розпізнавання зображень кількістю шарів згортки. Alexis Conneau [15] відзначає, що для обробки природних мов використовуються відносно неглибокі згорткові мережі, на відміну від ЗНМ для розпізнавання зображень, які містять набагато більше шарів згортки. Для порівняння архітектури для NLP часто складаються з 1-2 шарів і обмежуються 5-6 шарами згортки, тоді як для успішного розпізнавання зображень ЗНМ налічують від 19 до 152 шарів.

2.2.6 Дерева рішень

Дерева рішень забезпечують ієрархічну декомпозицію даних, для поділу яких використовується умова відносно значення атрибута [12]. Умова або предикат – це наявність або відсутність одного або декількох слів. Поділ простору даних здійснюється рекурсивно, поки вузли листів не містять певну мінімальну кількість термінів, які використовуються для класифікації.

Існують і інші типи предикатів, які залежать від подібності документів для співвіднесення наборів термінів, які можуть бути використані для подальшого розподілу. Виділяють декілька типів розбиття даних: розбиття за

одиничним атрибутом (single attribute split), який використовує наявність або відсутність конкретних слів або фраз на певному вузлі дерева для того, щоб здійснити розбиття [13]; багато атрибутивне розбиття на основі подібності (similarity-based multi-attribute split), яке використовує документи або часто вживані кластери слів та подібність документів до цих кластерів слів, щоб виконати розділення; багато атрибутивне розбиття на основі дискримінанта (discriminat-based multi-attribute split), який використовує такі дискримінанти, як наприклад дискримінант Фішера для розбиття даних [14].

2.4 Оцінка якості алгоритмів

Після розробки та навчання моделей необхідним кроком є оцінка їх ефективності. Типовою оцінкою якості класифікатора є точність, яку для бінарної задачі можна легко отримати з матриці помилок (рис. 2.2). Цей показник можна обчислити так:

$$Accuracy = (TP + TN) / (TP + FN + TN + FP), \quad (2.26)$$

де TP — вірно визначений позитивний клас; TN — вірно визначений негативний клас; FP – хибний позитивний клас; FN — хибний негативний клас.

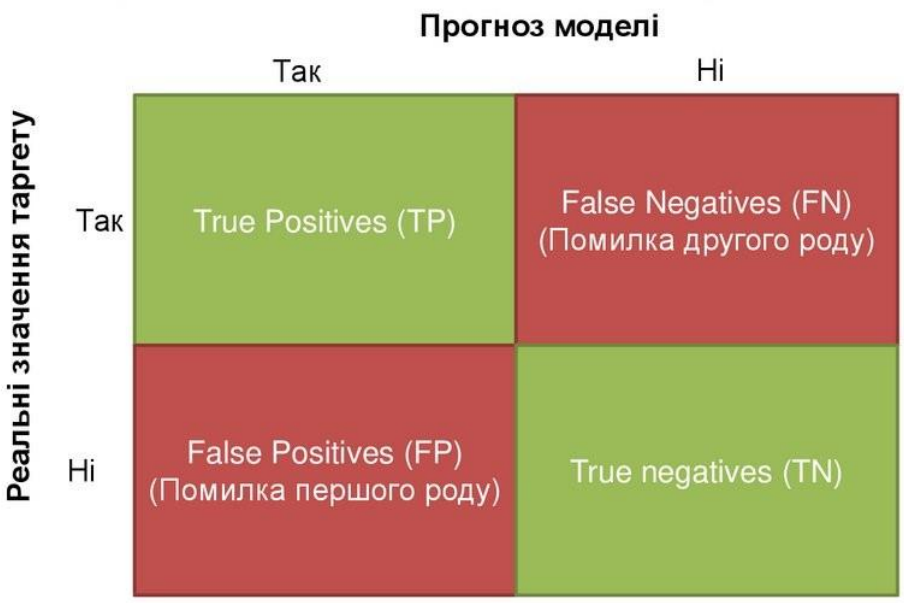

Рис 2.2. Матриця помилок

Емпіричні дані показують, що показник точності сильно залежать від сбалансованості даних: використання цього показника дає помилкові висновки, якщо класи є сильно дисбалансованими.

У випадку, коли дані незбалансовані доцільно перевірити, наскільки ефективно класифікатор класифікує лише частину даних, а саме позитивні або негативні класи даних. Прикладами таких метрик є чутливість (precision) та повнота (recall).

Чутливість доцільно використовувати, коли помилково позитивна класифікація небажана. Вона розраховується за такою формулою:

$$Precision = TP / (TP + FP) \quad (2.27)$$

Метрику повноти використовують, коли треба уникнути помилково негативної класифікації. Її обчислюють за формулою:

$$Recall = TP / (TP + FN) \quad (2.28)$$

Також є показним, який є гармонічним середнім двох попередніх оцінок – F-міра. Це загальна міра точності моделі, яка поєднує в собі чутливість та повноту. Тобто, хороший показник F_1 означає, що у вас низька кількість хибних позитивних та хибних негативних класифікацій. Міра F_1 вважається ідеальною, якщо вона дорівнює 1, і катастрофічною, коли вона дорівнює 0.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (2.28)$$

Висновки до розділу

У розділі було описано підходи до аналізу тональності тексту, способи представлення тексту у числовому вигляді, а саме Bag-of-Words, який представляє речення як список слів, які з'являються у ньому і їх частоти, TF-IDF векторизатор, в якому кожне слово представляє частоту з якою воно вживається в реченні збалансовану частотою цього слова у всьому корпусі текстів, та Word2Vec векторизатор, який представляє кожне слово у просторі векторів. Також було описано алгоритми машинного навчання, такі як Наївний Байєсівський класифікатор, Байєсові мережі, метод максимальної ентропії, метод опорних векторів, згорткова нейронна мережа та дерева рішень.

Зробивши огляд алгоритмів, які використовуються для аналізу тональності тексту, було прийнято рішення використовувати в дослідженні такі алгоритми, як наївний Байєсівський класифікатор, метод опорних векторів та згорткову нейронну мережу.

Також важливим завданням є оцінка роботи моделі, тому в розділі було представлено критерії для оцінки побудованої моделі: точність, повнота та F-міра.

РОЗДІЛ 3. ОПИС ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Наше дослідження полягає в аналізі емоційного навантаження тексту коментарів з соціальних мереж в період передвиборчих перегонів на пост Президента України в 2019 році.

Як відомо, коментарі в соціальних мережах є прямим віддзеркаленям громадської думки, тому їх аналіз на предмет сентименту може передбачити результат виборів.

У цьому дослідженні ми використовуватимемо машинне навчання, оскільки підхід з використанням словників вважається застарілим і не ефективним, а методи МН дають більш точні результати і простіші у реалізації.

Для проведення дослідження було зібрано коментарі під публікаціями кандидатів у Президенти України 2019 В.О.Зеленського та П.О.Порошенка за два місяця до першого туру президентських виборів. Вся вибірка складається з близько 20 тис. записів, по 10 для кожного з кандидатів.

Для кожного кандидата ми навчимо окрему модель. Потім застосуємо дану модель для класифікації коментарів до публікацій періода першого та другого турів виборів в хронологічному порядку і з отриманих результатів дослідимо зміну громадської думки в залежності від тодішніх подій.

Для визначення тональності тексту буде використано три алгоритма: наївний Байєсівський класифікатор, метод опорних векторів та згорткова нейронна мережа.

Для того, щоб зробити дані придатними для алгоритмів машинного навчання, їх необхідно перетворити у вектори. Для векторизації тексту ми застосуємо два алгоритми: мішок слів та TF-IDF векторайзер; і порівняємо їх ефективність.

Для розробки програмного продукту в рамках магістерської дисертації було обрано мову програмування Python, оскільки ця мова найкраще підходить для машинного навчання та проектів на основі штучного інтелекту. Python

надає доступ до потужних бібліотек та фреймворків для машинного навчання (ML), він простий для розуміння та не залежить від платформи.

3.1. Аналіз та попередня обробка даних

Для зібраних даних було розставлено мітки класів: 0 — негативний сентимент, 1 — позитивний. Навчальна вибірка містить наступні поля: автор, дата, коментар та сентимент. Приклад частини даних наведено на рис. 3.1.

text	sentiment
Лучший , Вы будущие нашей страны)	1.0
UAUAUAUA🤘🤘🤘UAUAUAUA	1.0
Мы за тебя👍	1.0
Надеяться нужно на себя и на свой голос ! ! ! А,то...	1.0
Тримайтеся і ми всіх переможемо!!!!!	1.0
Слушаешь, закрываешь глаза и слышишь Юлю Тимош...	0.0
👏 красиво сказано	1.0
Молодець, Володимире!	1.0
@vishna_art вибач що розбив твої рожеві окуляри!	0.0
став лайк,якщо ти за Зеленського	1.0

Рис. 3.1. Зразок даних

Будемо вирішувати задачу бінарної класифікації, оскільки було зроблено припущення, що люди які залишають коментарі не є політично нейтральними, тому нейтральних коментарів або зовсім не буде, або буде зовсім незначна частка, якою можна знехтувати.

Класи є незбалансованими. Для В. Зеленського позитивний клас складає 83%, а для П. Порошенка — 38%. Про це необхідно пам’ятати під час навчання моделі.

Перш ніж почати передобробку даних, необхідно їх проаналізувати. Оскільки ми будуємо окремі моделі для кожного з кандидатів, обчислимо наступні основні статистичні показники текстової змінної (коментаря) для кожного з них:

- кількість слів у коментарі;
- посилання на аккаунти інших користувачів (починаються з @);
- кількість тегів (починаються з #);
- кількість слів, написаних заголовними літерами. Інколи використовується, щоб висловити сильні (негативні) емоції;
- кількість знаків питання та знаків оклику;
- кількість посилань у коментарі (починаються з http(s));
- кількість емотиконів.

Загалом, кількість слів в більшості коментарів, досить низька — до 25 слів. Є коментарі, що містять лише одне слово або емотикон. Також можна відзначити, що позитивні коментарі у В. Зеленського містять більше слів, ніж негативні. Наведемо нижче розподіл кількості слів у коментарі за класами для кожного з кандидатів (рис. 3.2, 3.3).

Більшість коментарів не містять згадок. Немає різниці в кількості згадок щодо настроїв.

Більшість коментарів не містять хеш-тегів. Тож ця змінна не буде збережена під час навчання моделей. Знову ж таки, немає різниці в кількості хеш-тегів щодо настроїв.

Більшість коментарів не містять великих літер, і ми не бачимо суттєвої різниці в розподілі між настроями.

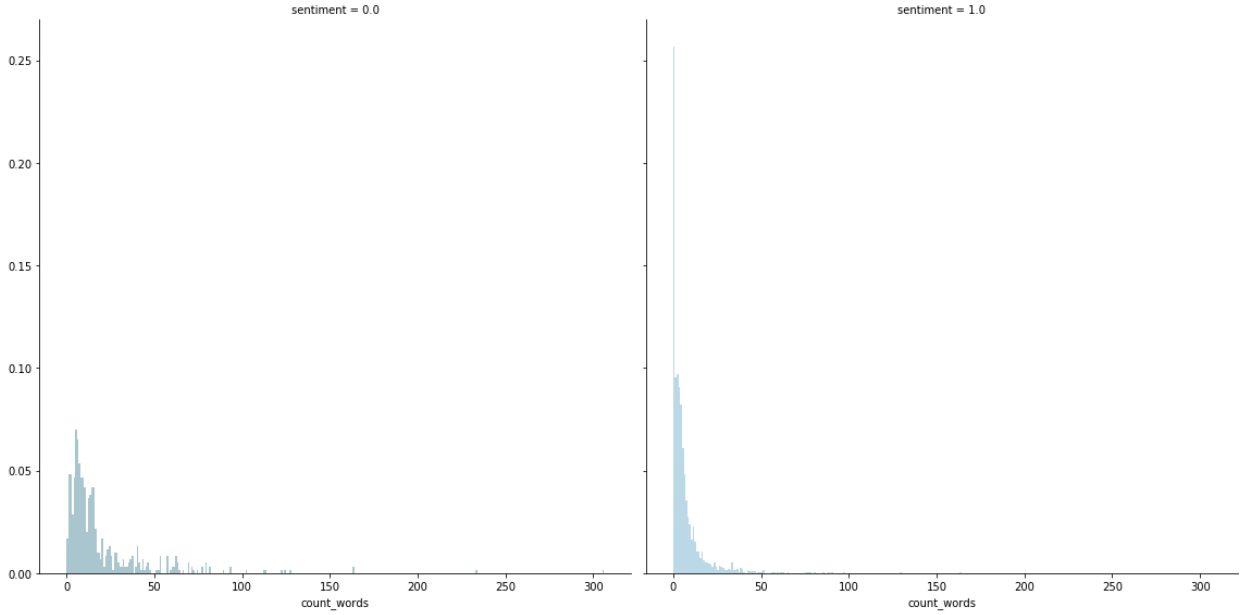

Рис. 3.2. Розподіл кількості слів у коментарі за класами для В.Зеленського

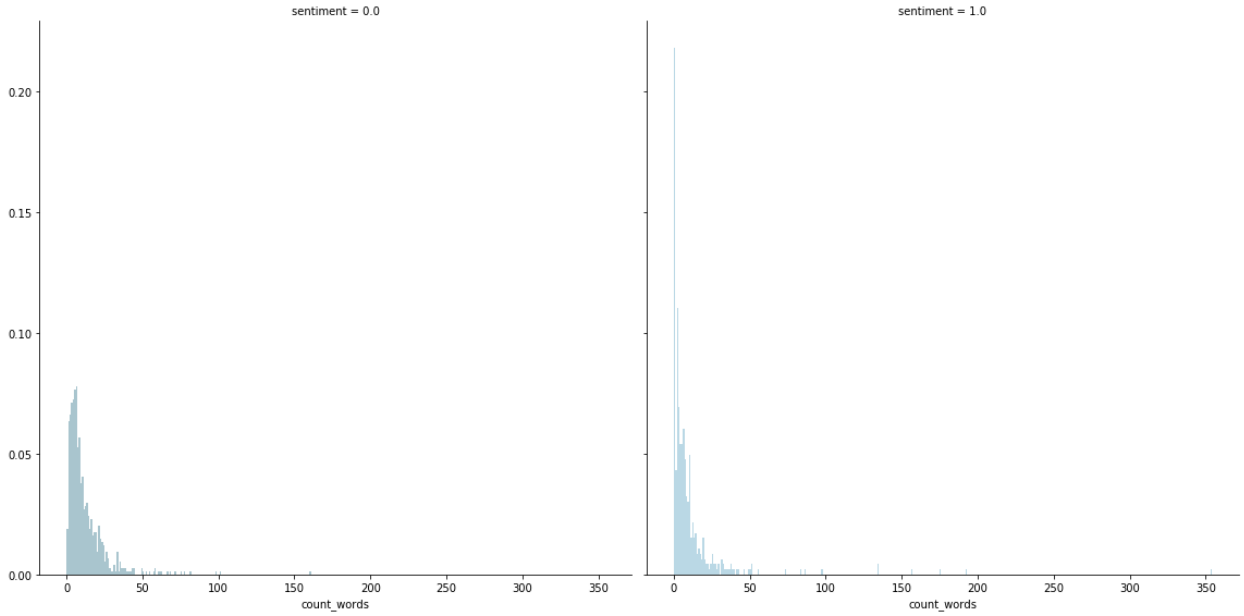

Рис. 3.3. Розподіл кількості слів у коментарі за класами для П.Порошенка

Негативні коментарі використовують дещо більше знаків питань чи знаків оклику, але знову ж таки різниця не суттєва.

Майже у всіх коментарях відсутні URL-адреси.

У позитивних коментарях емотикони зустрічаються набагато частіше (рис. 3.4, 3.5).

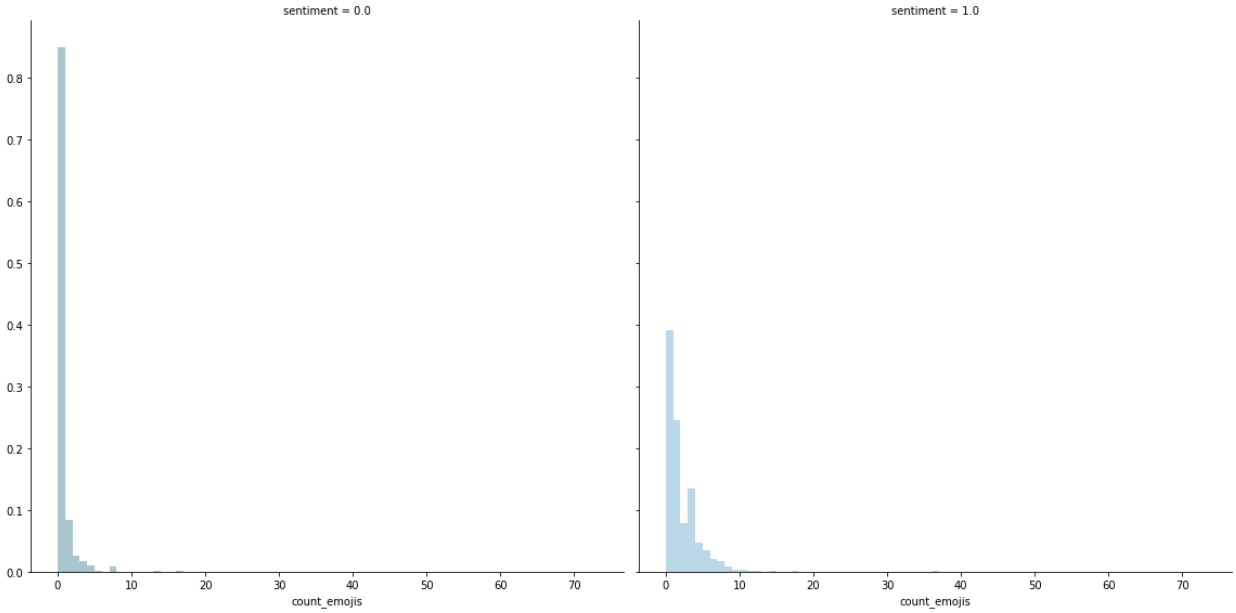

Рис.3.4 Розподіл кількості емоджі у коментарі за класами для В.Зеленського

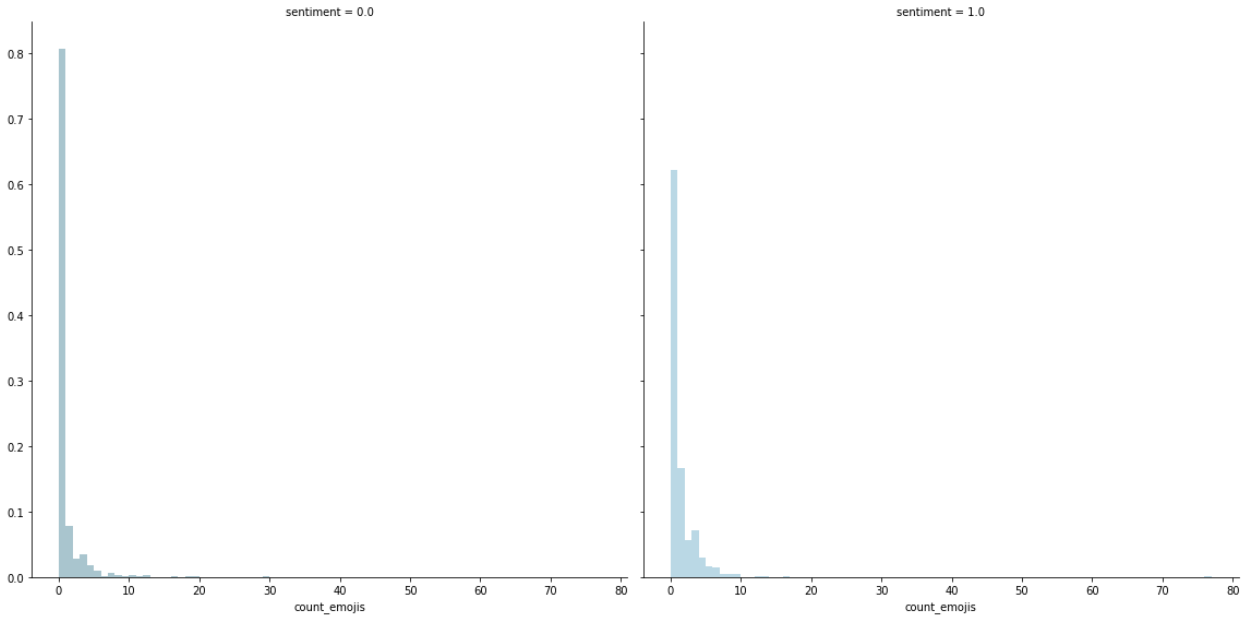

Рис. 3.5. Розподіл кількості емоджі у коментарі за класами для П.Порошенка

Перш ніж почати використовувати текст коментарів, його потрібно очистити від непотрібної інформації, а саме:

- видалити згадки, оскільки вони не несуть емоційного навантаження;
- видалити знак хештега, але не сам хештег, оскільки він може містити інформацію;
- перевести всі слова до нижнього регістру;
- видалити всі розділові знаки, включаючи знаки запитання та знаки оклику;
- видалити URL-адреси, оскільки вони не містять корисної інформації;
- конвертувати емоджі в одне слово;
- видалити цифри;
- видалити стоп-слова;
- застосувати стемінг, щоб зберегти основу слова без закінчення чи суфіксів.

Оскільки коментарі написані українською та російською мовами, ми видалятимемо російські і українські стоп-слова та застосовуватимемо два стемера: спочатку російський, потім український. Приклад коментарів наведено на рис. 3.6.

	text	count_words	count_mentions	count_hashtags	count_capital_words	count_excl_quest_marks	count_urls	count_emojis
0	fire	0	0	0	0	0	0	1
2	владимир президент smilingfacewithsmilingey uk...	3	0	0	0	3	0	1
4	посад grimacingfac	6	0	0	0	2	0	1
6	крас redheart fire	1	0	0	0	1	0	2
8	зе став лайк	5	0	0	0	1	0	0

Рис. 3.6. Приклад коментарів після обробки

Якщо після такої очистки з’являться коментарі, що не містять жодного слова, їх буде видалено, оскільки вони не містять інформації про сентимент.

Тепер, коли текст коментарів очищено, ми можемо дізнатись, які слова найчастіше зустрічаються в коментарях. Нижче показано найбільш часто вживані 15 слів під публікаціями кожного з кандидатів у Президенти (рис.3.7-3.8).

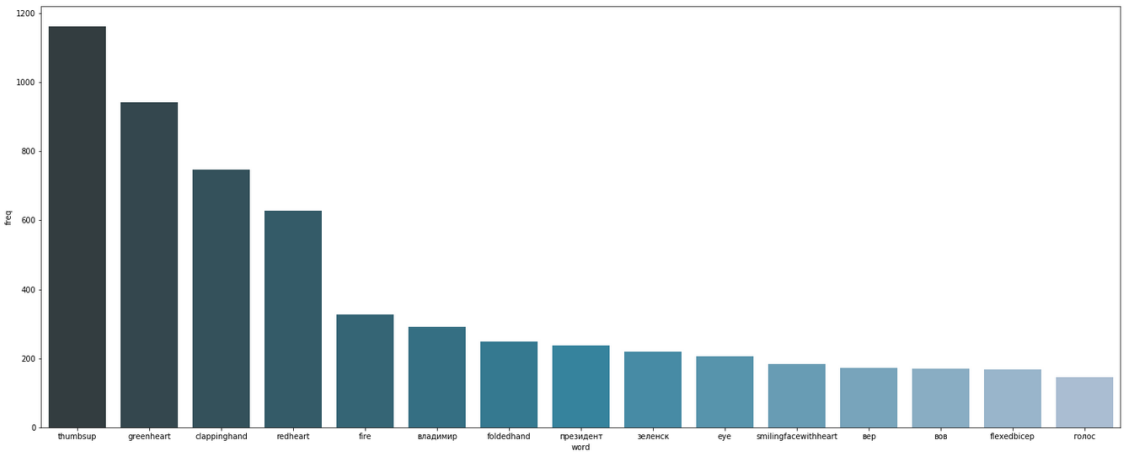

Рис. 3.7. П’ятнадцять найчастіше вживаних слів під публікаціями В.Зеленського

Під публікаціями В.Зеленського найчастіше вживалися емотикони *thumbsup, greenheart, clappinghand*; зі слів найпопулярнішими виявились *Владимир, президент, Зеленский, верю*.

Що стосується П.Порошенка, найбільш вживаними словами виявились *Президент, Порошенко, Украина*; емоджі – *thumbsup*, *facewithtearsofjoy*, *Ukraine* (рис. 3.8).

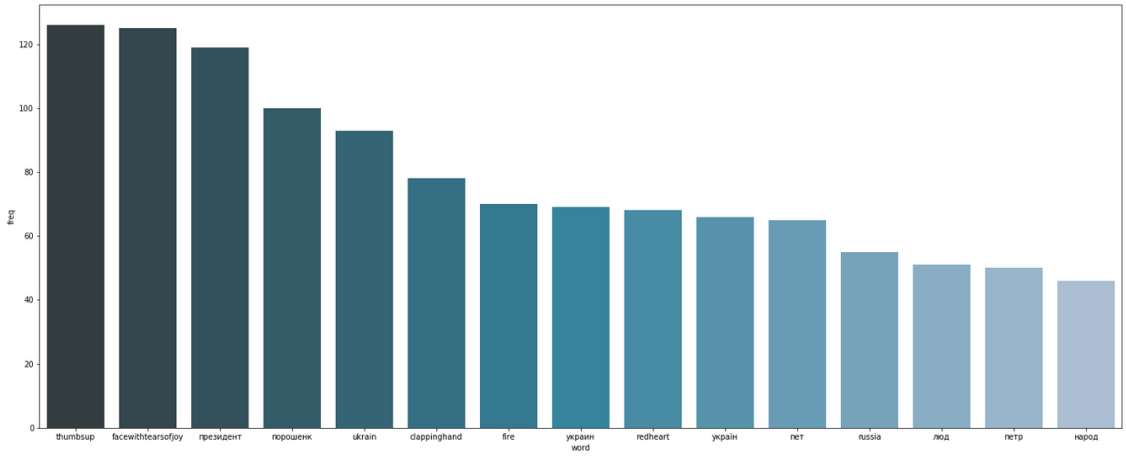

Рис. 3.8. П’ятнадцять найчастіше вживаних слів під публікаціями П.Порошенка

3.2 Налаштування гіперпараметрів та кросвалідація

Як ми побачимо нижче, і векторайзери і класифікатори мають параметри, що піддаються налаштуванню. Щоб підібрати найкращі параметри, потрібно виконувати перевірку на окремому тестоваму наборі, який не використовувався під час навчання моделі. Однак використання лише одного тестового набору може не дати достовірних результатів. Оскільки дані діляться випадковим чином на тренувальний та перевірочний сет, модель може показати хорошу точність на одному тестовому наборі, але на інших даних результати можуть дуже відрізнятись. Тому для отримання більш точної оцінки ми будемо використовувати перехресну перевірку (кросвалідацію).

Кросвалідація — це техніка валідації якості моделі на незалежних даних. При цьому дані розбиваються на *n* частин, на *n-1* частинах модель навчається (тренувальний набір даних), а на останній частині (тестовий набір) виконується перевірка. Цей процес повторюється *n* разів, в результаті чого кожна з частин в певний момент виступає як тестовий набір даних, а результати валідації усереднюють по всім циклам (фолдам). Метою кросвалідації є оцінка очікуваного рівня відповідності моделі даним незалежним від тих, на яких

модель тренувалась. Використовуючи кросвалідацію ми визначаємо оптимальні параметри моделі з оглядом на метрики чутливості (precision) та повноти (recall).

3.3 Метрики оцінки якості моделі

Для оцінки якості побудованої моделі не доцільно використовувати метрику точності, оскільки класи даних є незбалансованими. Більш доречним буде використовувати показники F-міри, яка є середнім гармонійним чутливості (recall) та повноти (precision).

3.4. Застосування методів МН для сентимент аналізу

Ми виконуватимемо навчання та тестування моделі шляхом кросвалідації з кількістю фолдів 5. Після навчання та перевірки точності ми виберемо 5 найкращих моделей за середньою помилкою кросвалідації. Для цих п’яти моделей побудуємо коробковий графік (boxplot) та оберемо найкращу модель, яка має найвищу середню точність і якомога меншу дисперсію.

3.4.1. Наївний Байєсівський класифікатор

Перед навчанням моделі задамо параметри для векторайзера: мінімальний та максимальний порог відсікання термінів (якщо слово зустрічається більше або менше разів від заданого порогу, воно не включається до словника) та

кількість n-грам. Також задамо параметр регуляризації для наївного Байєсівського класифікатора. За допомогою кросвалідації будуть обрані найбільш оптимальні з цих параметрів.

Задамо наступний набір параметрів для кросвалідації (рис 3.9):

```
parameters = {
    'features__pipe__vect__max_df': (0.1, 0.25, 0.5, 0.75, 0.85, 1.),
    'features__pipe__vect__ngram_range': ((1, 1), (1, 2), (1, 3), (1, 4)),
    'features__pipe__vect__min_df': [1],
    'clf__alpha': (0.1, 0.25, 0.5, 0.75, 1)
}
```

Рис. 3.9. Набір параметрів для моделі НБК

При використанні векторизатора «мішок слів», оптимальним набором параметів для моделей обох кандидатів виявились: регуляризація — 0.1. Оптимальний максимальний порог відсікання склав 1.0 для В. Зеленського та 0.85 для П. Порошенка, кількість n-грам — (1, 2) для В. Зеленського та (1, 1) для П. Порошенка. При таких параметрах точність моделі за метрикою F1 складає для В.Зеленського 94%, а для П.Порошенка – 90%. Результати підбору оптимальних параметрів для НБК представлені на рис. 3.10 та 3.12, а коробкові графіки точності на рис. 3.11 та 3.13.

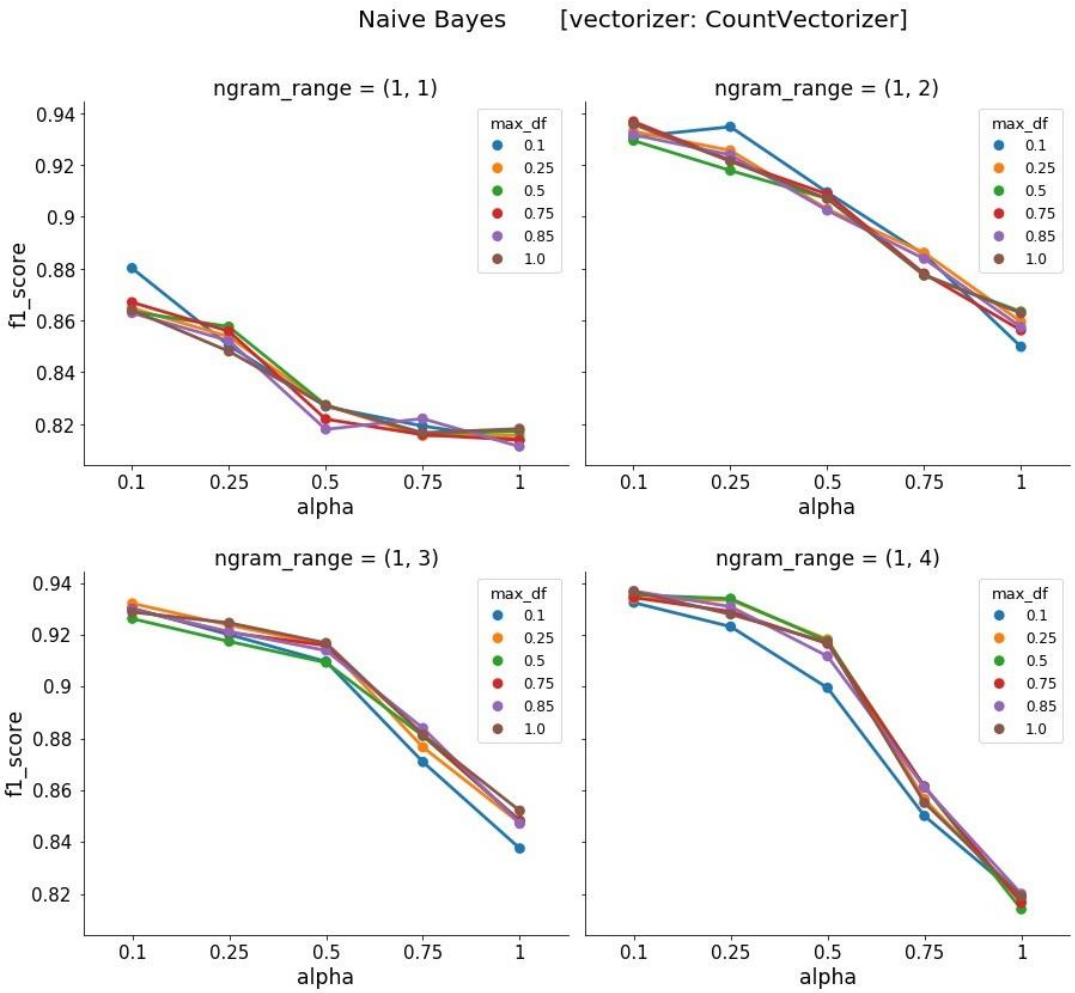

Рис. 3.10. Результати підбору параметрів для НБК на вибірці В.Зеленського (векторизатор Bag-of-Words)

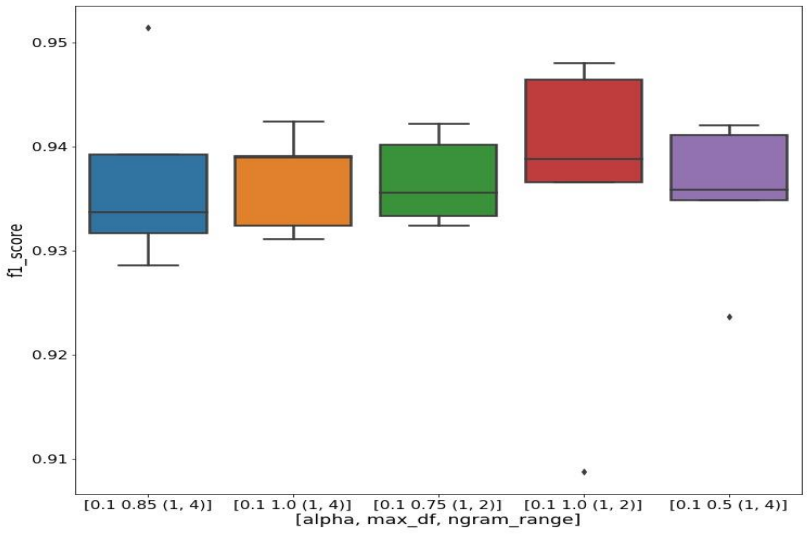

Рис. 3.11. Boxplot для НБК з векторайзером Bag-of-Words (В. Зеленський)

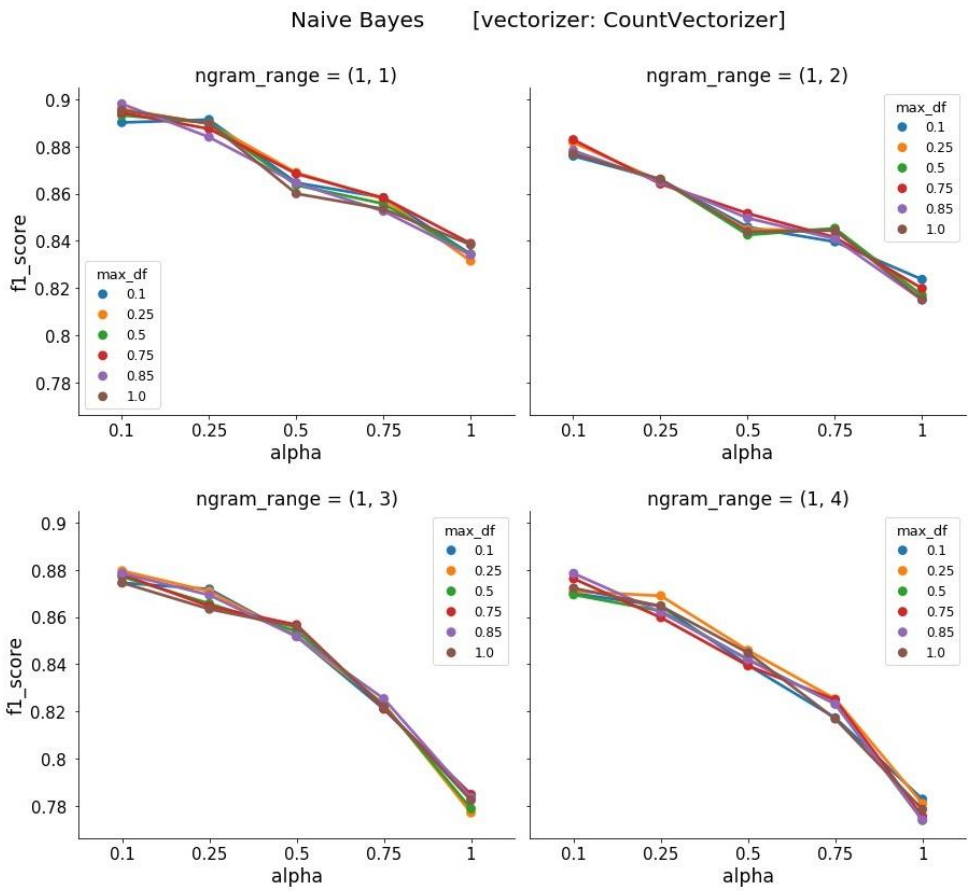

Рис. 3.12. Результати підбору параметрів для НБК на вибірці П.Порошенка (векторизатор Bag-of-Words)

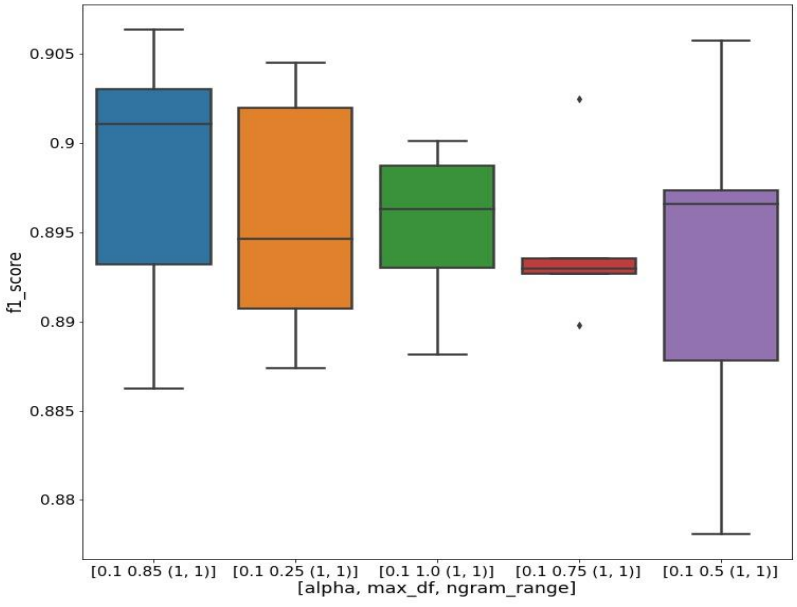

Рис. 3.13. Boxplot для НБК з векторайзером Bag-of-Words (П. Порошенко)

При використанні TF-IDF векторизатора, оптимальним набором параметів для моделі, що навчалася на даних зі сторінки В. Зеленського виявились: регуляризація — 0.1, кількість n-грам — (1, 4), максимальний поріг відсіву — 0.75. При цьому точність моделі складає 94.2%. Для даних зі сторінки П. Порошенка оптимальними параметрами є: регуляризація — 0.1, кількість n-грам — (1, 1), максимальний поріг відсіву — 0.1. При таких параметрах точність моделі складає 92%. Результати підбору оптимальних параметрів для НБК з TF-IDF векторизатором представлені на рис. 3.14 та 3.16, а коробкові графіки точності на рис. 3.15 та 3.17.

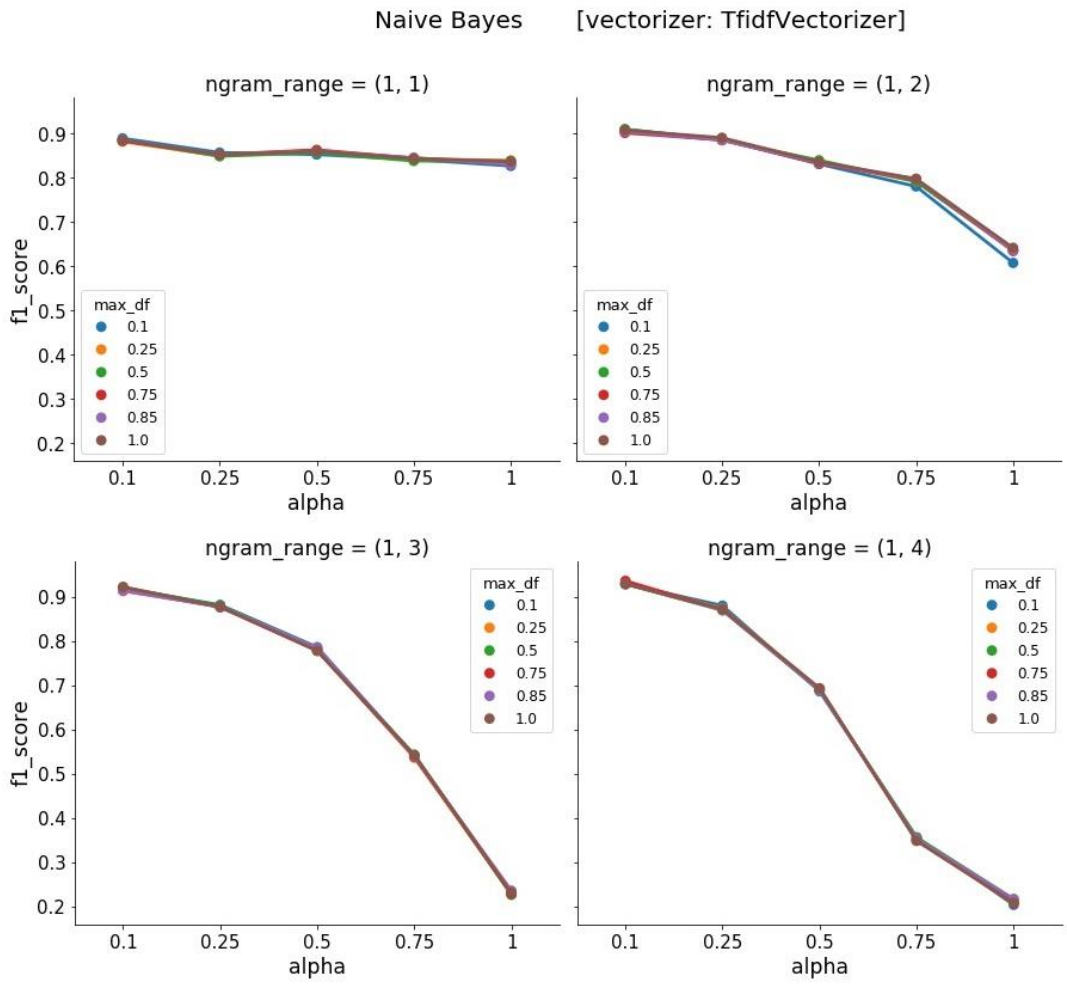

Рис. 3.14. Результати підбору параметрів для НБК на вибірці В.Зеленського (TF-IDF векторайзер)

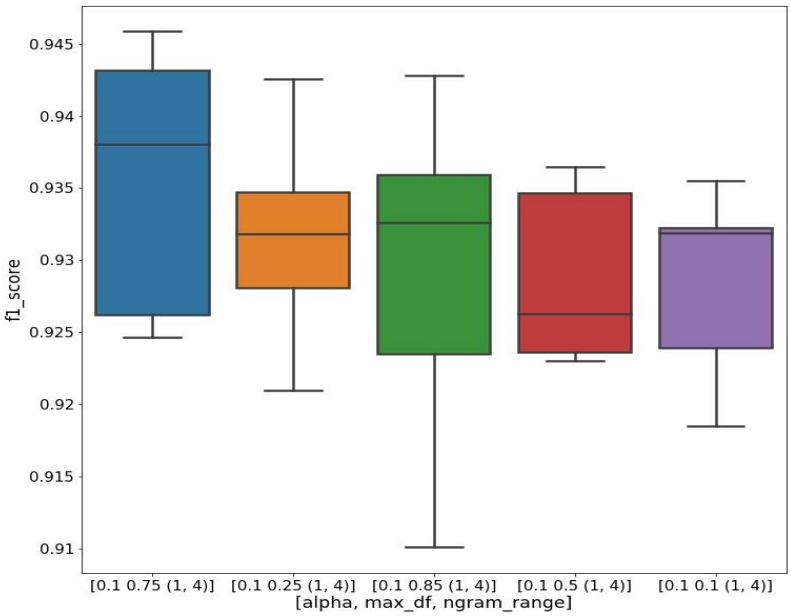

Рис. 3.15. Boxplot для НБК з векторизатором TF-IDF (В. Зеленський)

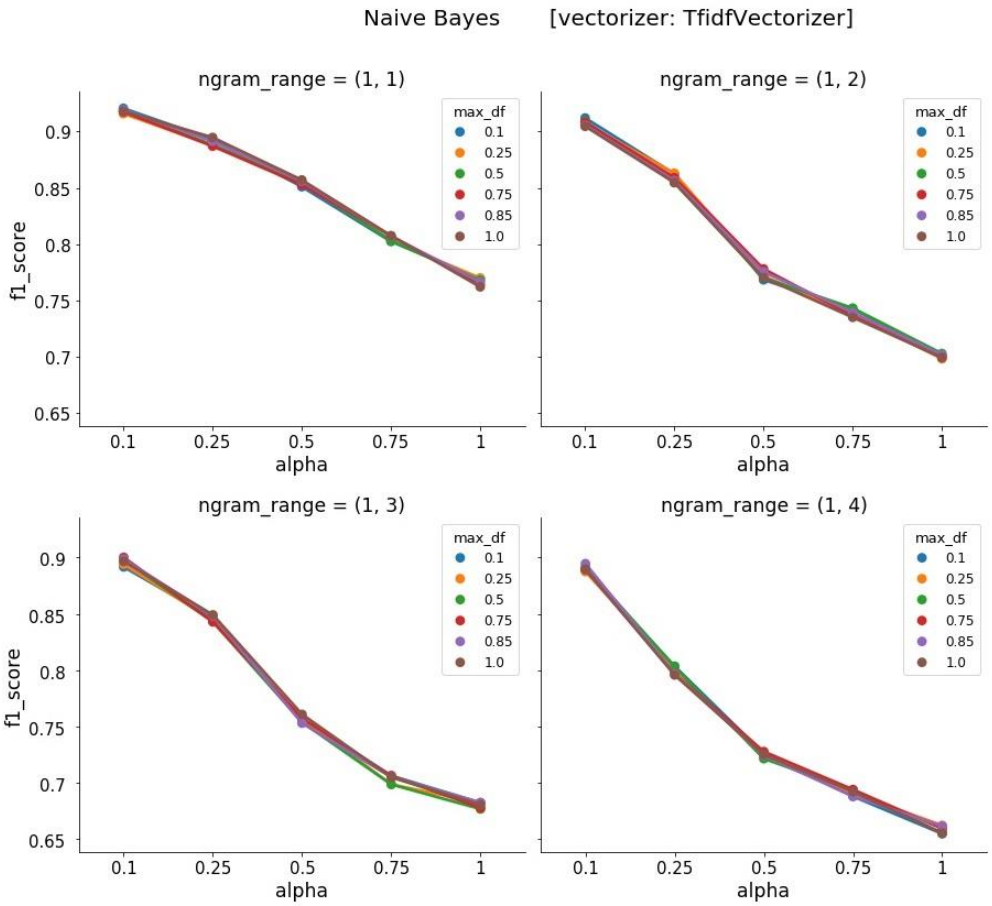

Рис. 3.16. Результати підбору параметрів для НБК на вибірці П.Порошенка (TF-IDF векторайзер)

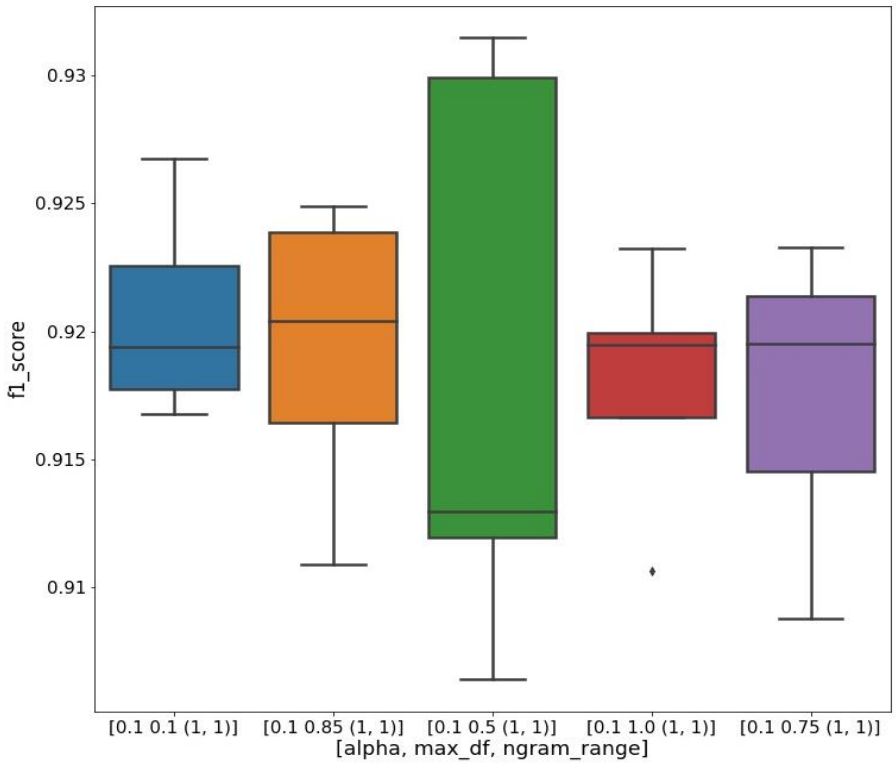

Рис. 3.17. Boxplot для НБК з векторизатором TF-IDF (П. Порошенко)

3.4.2. Метод опорних векторів

Задамо наступний набір параметрів для кросвалідації (рис. 3.13):

```
parameters = {
    'features__pipe__vect__max_df': (0.1, 0.25, 0.5, 0.75, 0.85, 1.),
    'features__pipe__vect__ngram_range': ((1, 1), (1, 2), (1, 3), (1, 4)),
    'features__pipe__vect__min_df': [1],
    'clf__C': (0.1, 0.25, 0.5, 0.75, 1),
    'clf__penalty': ['l1'],
    'clf__dual': [False]
}
```

Рис. 3.18. Параметри для методу опорних векторів

При використанні векторизатора «мішок слів», оптимальним набором параметрів для моделей обох кандидатів виявились: регуляризація — 0.75. Оптимальна кількість n-грам для вибірки В. Зеленського складає (1, 3), для П. Порошенка – (1, 1); максимальний поріг відсікання 0.75 – для В. Зеленського, 0.5 – для П. Порошенка. При таких параметрах точність моделі складає для В. Зеленського 94.5%, а для П.Порошенка — 90%. Результати підбору оптимальних параметрів для SVM представлені на рис. 3.19 та 3.21, а коробкові графіки точності на рис. 3.20 та 3.22.

При використанні векторизатора TF-IDF для моделі, побудованій на даних зі сторінки В.Зеленського найкращими параметрами є: регуляризація — 1, максимальний поріг відсікання — 0.5, кількість n-грам — (1, 1). А для П.Порошенка: регуляризація — 0.75, максимальний поріг відсікання — 0.75, кількість n-грам — (1, 3). При таких параметрах точність моделі складає для В.Зеленського 93%, а для П.Порошенка — 94%. Результати підбору оптимальних параметрів для SVM з TF-IDF векторайзером представлені на рис. 3.23 та 3.25, а коробкові графіки точності на рис. 3.24 та 3.26.

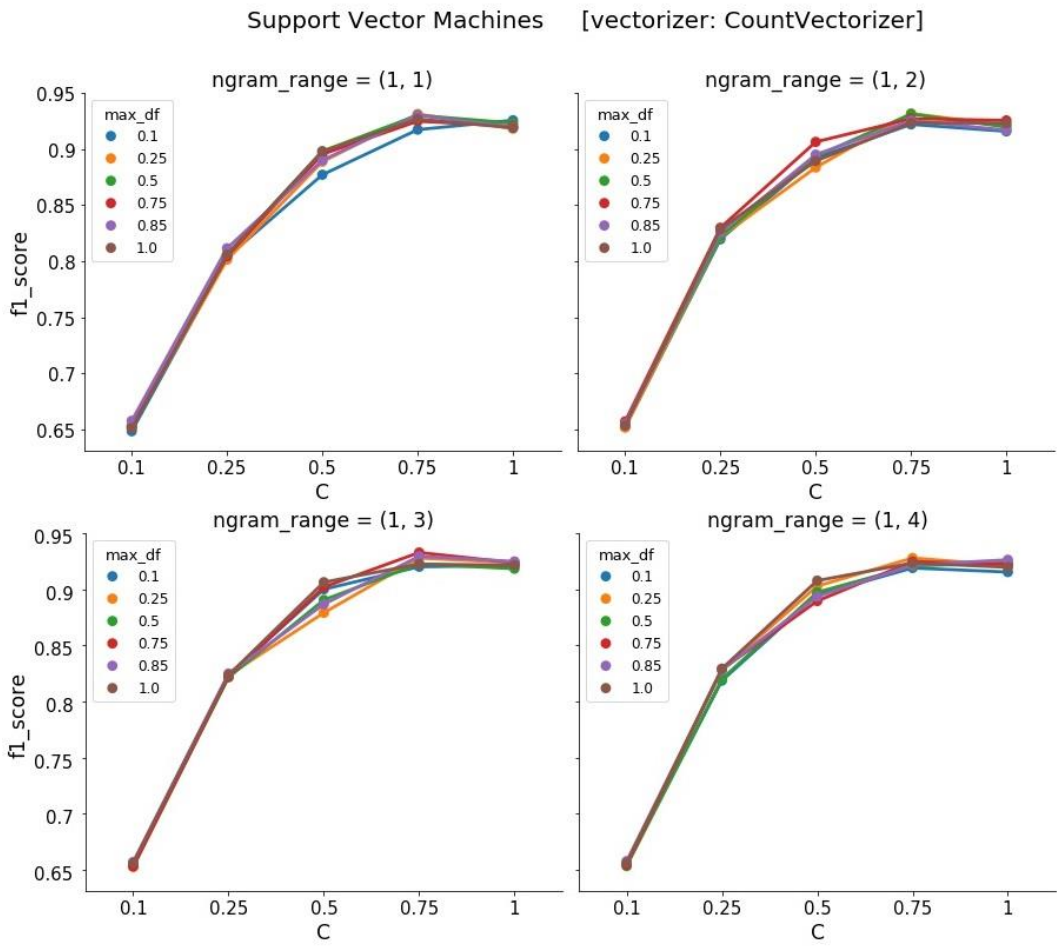

Рис. 3.19. Результати підбору параметрів для SVM на вибірці В. Зеленського (векторизатор Bag-of-Words)

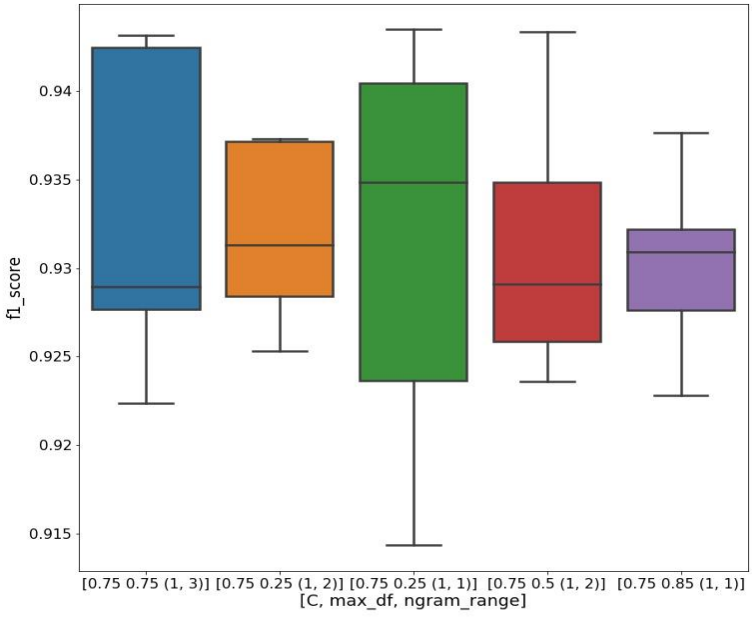

Рис. 3.20. Boxplot для SVM з векторайзером Bag-of-Words (В. Зеленський)

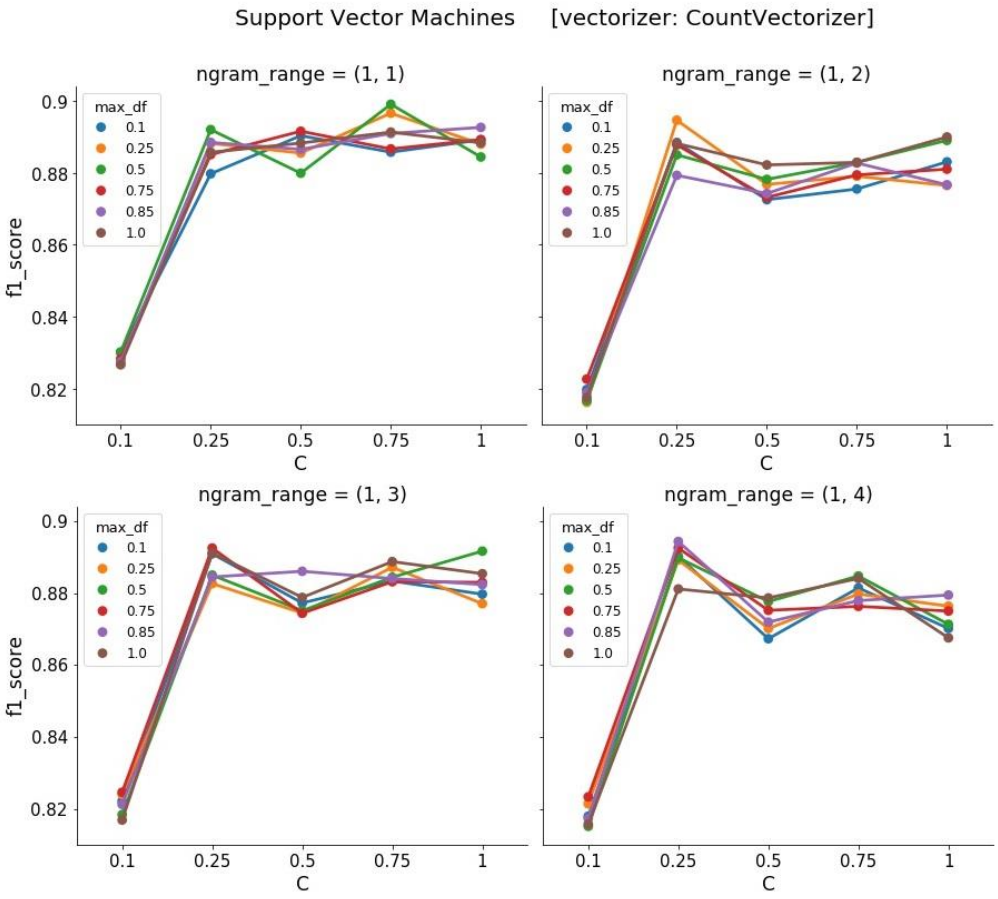

Рис. 3.21. Результати підбору параметрів для SVM на вибірці П.Порошенка (векторизатор Bag-of-Words)

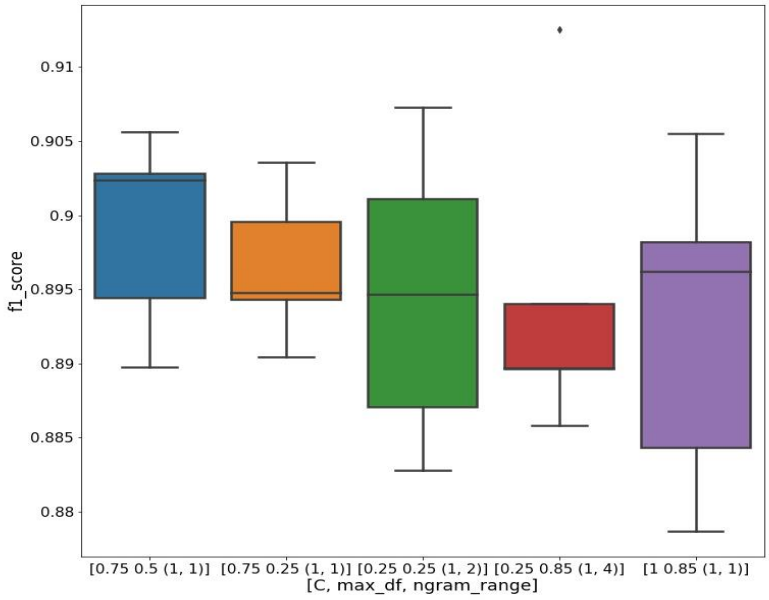

Рис.3.22. Boxplot для SVM з векторайзером Bag-of-Words (П. Порошенко)

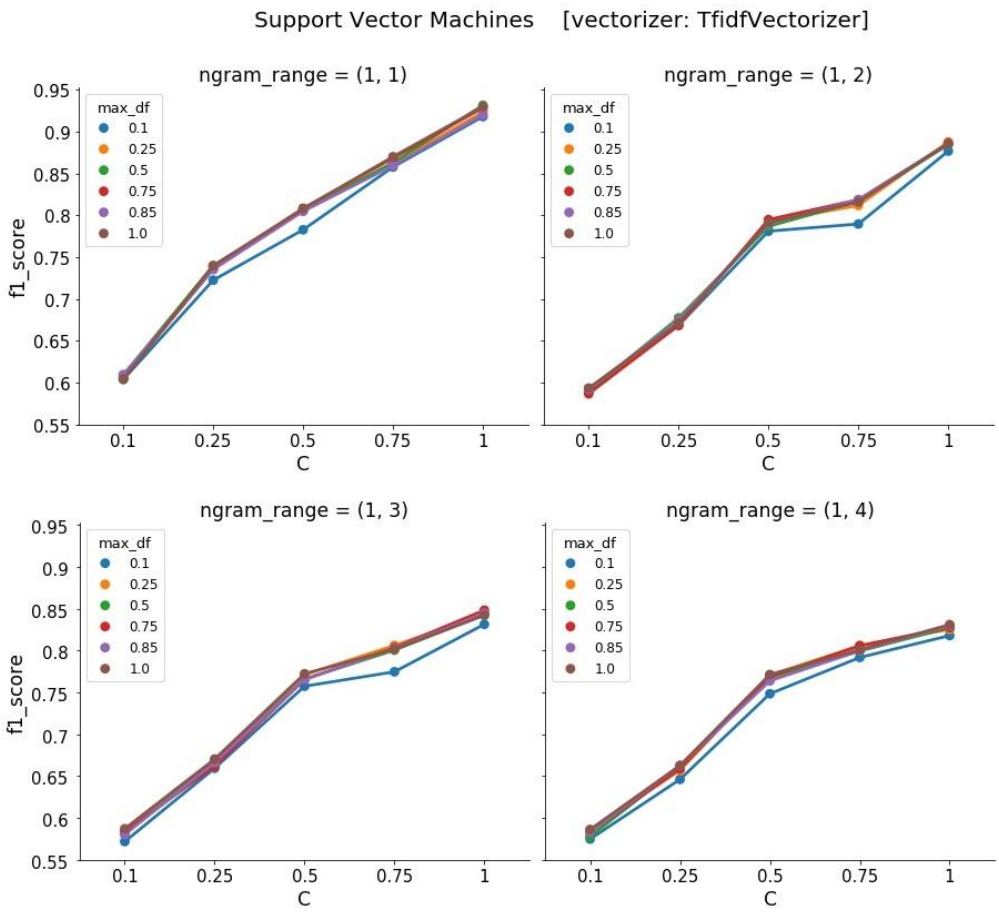

Рис. 3.23. Результати підбору параметрів для SVM на вибірці В.Зеленського (TF-IDF векторайзер)

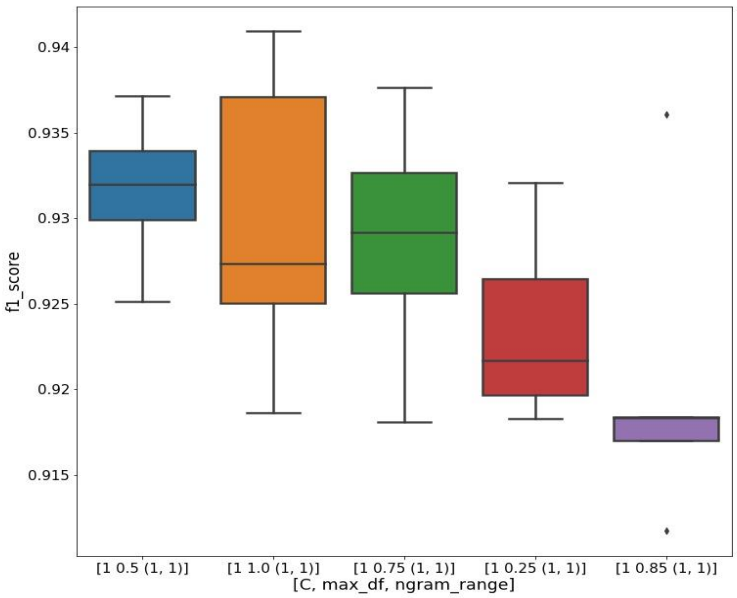

Рис. 3.24. Boxplot для SVM з векторайзером TF-IDF (В. Зеленський)

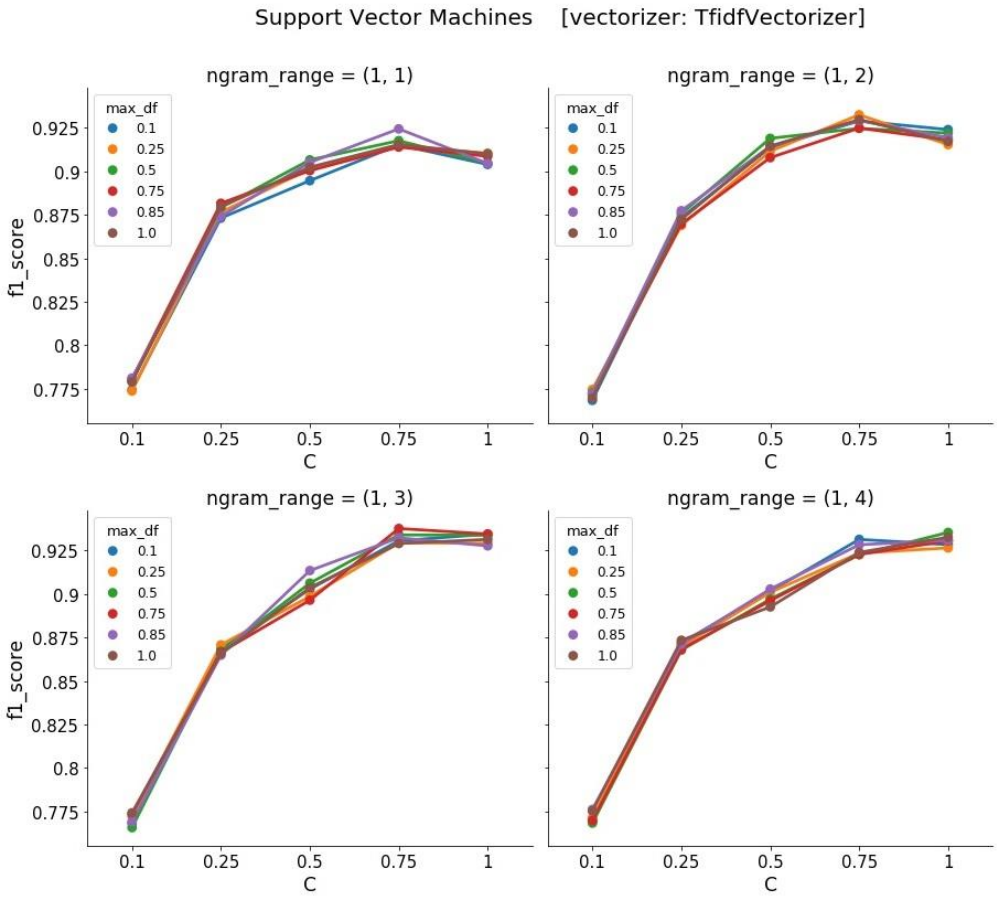

Рис. 3.25. Результати підбору параметрів для SVM на вибірці П. Порошенка (TF-IDF векторайзер)

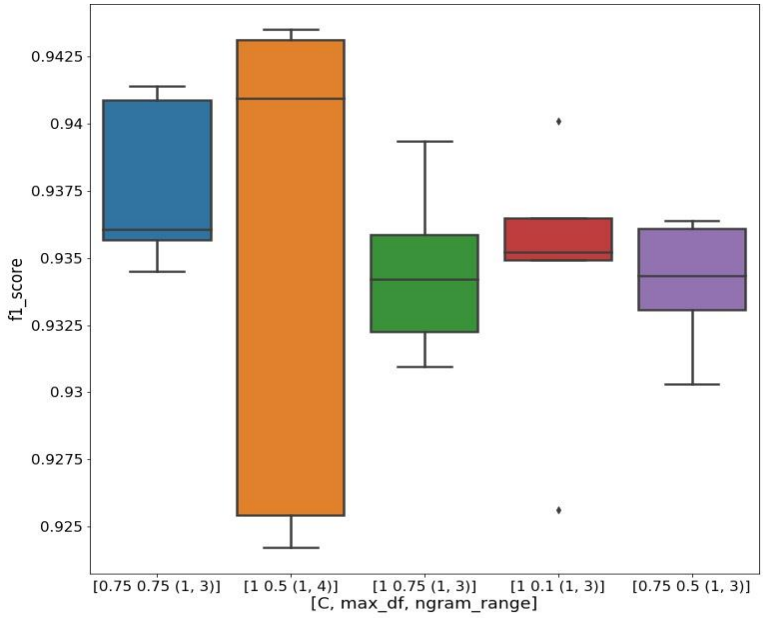

Рис. 3.26. Boxplot для SVM з векторайзером TF-IDF (П. Порошенко)

3.4.3 Згорткова нейронна мережа

Для згорткової нейронної мережі було задано наступні параметри:

- Функція активації: ReLU
- Регуляризація (L2): 3
- Дропаут (0.4).
- Розмір батча: 100

Шляхом кросвалідації ми будемо підбирати кількість шарів згортки (1, 2, 3), розмір ядра згортки (3, 4, 5) та кількість фільтрів (100, 200, 300). Параметри наведено на рис. 3.18.

```
parameters = {
    'clf__layers': (1, 2, 3),
    'clf__number_of_filters': (100, 200, 300),
    'clf__kernel': (3, 4, 5)
}
```

Рис. 3.27. Набір параметрів для ЗНМ

Для згорткової нейронної мережі слова було переведено у векторне представлення шляхом використання алгоритма Word2Vec (Skip-Gram). Оптимальним набором параметрів для моделей обох кандидатів виявились: кількість фільтрів — 300, кількість шарів згортки — 1. Розмір ядра згортки, що дає найкращу точність для даних В.Зеленського становить 4, а для П.Порошенка — 3. При таких параметрах точність моделі складає для В.Зеленського 95.6%, а для П.Порошенка — 95.5%. Результати підбору оптимальних параметрів для ЗНМ представлені на рис. 3.28 та 3.30, а коробкові графіки точності на рис. 3.29 та 3.31.

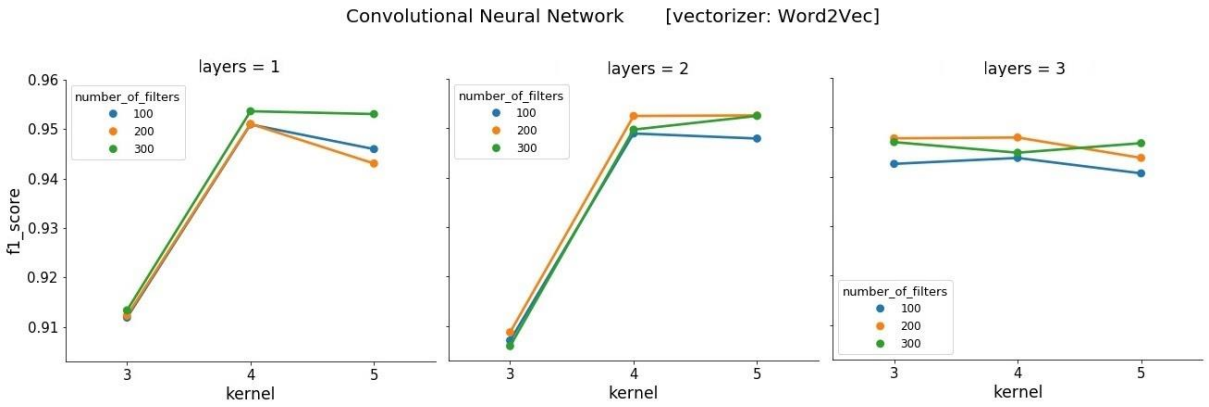

Рис. 3.28. Результати підбору параметрів для ЗНМ на вибірці В. Зеленського

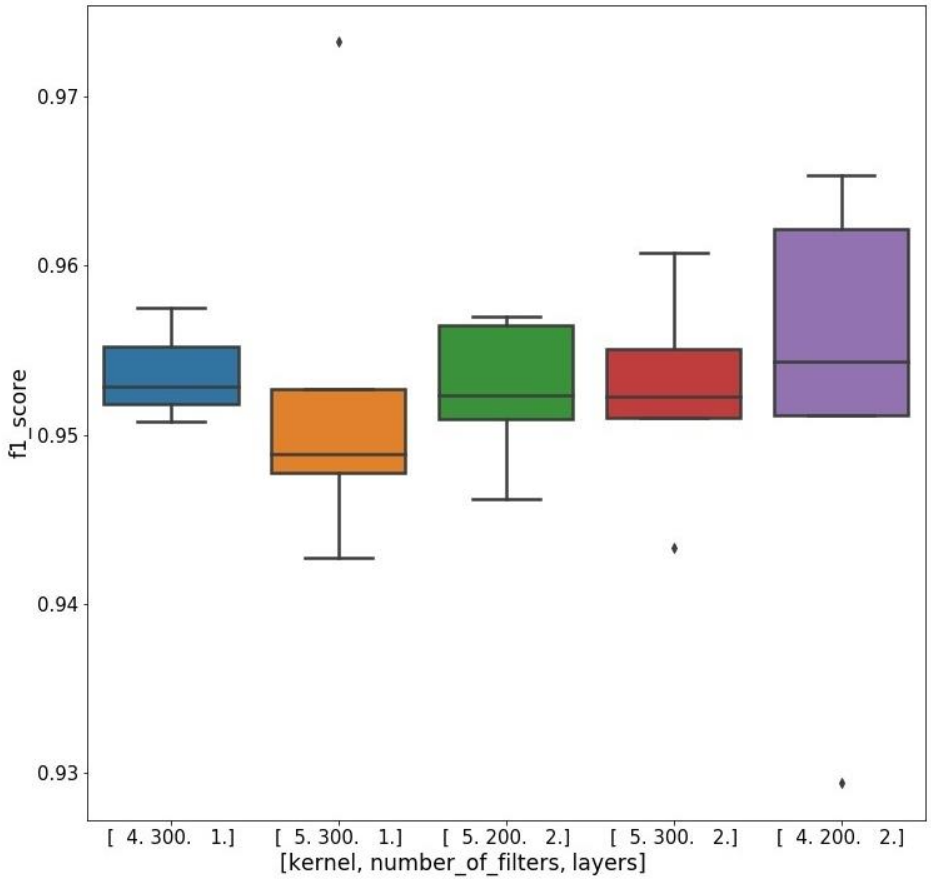

Рис. 3.29. Boxplot для ЗНМ (В. Зеленський)

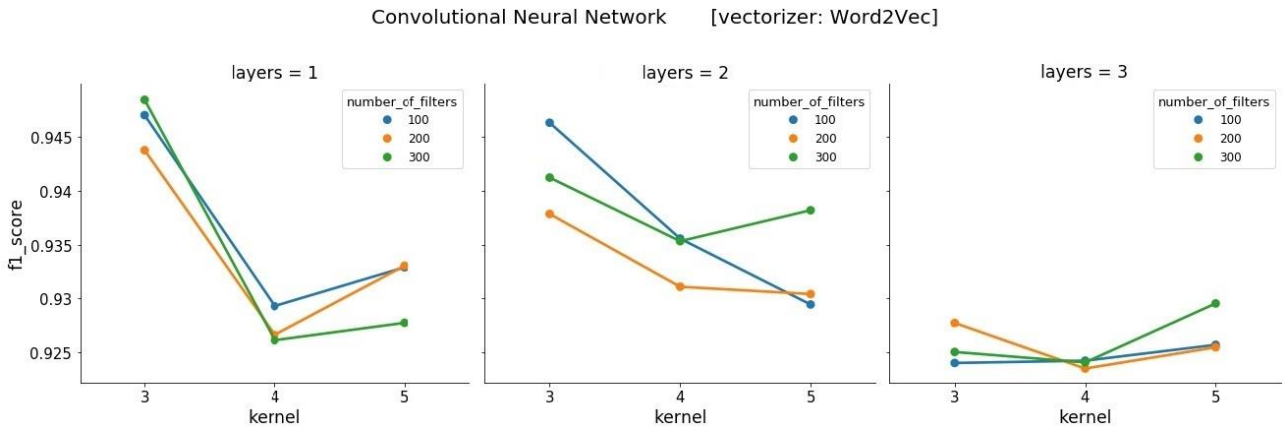

Рис. 3.30. Результати підбору параметрів для ЗНМ на вибірці П. Порошенка

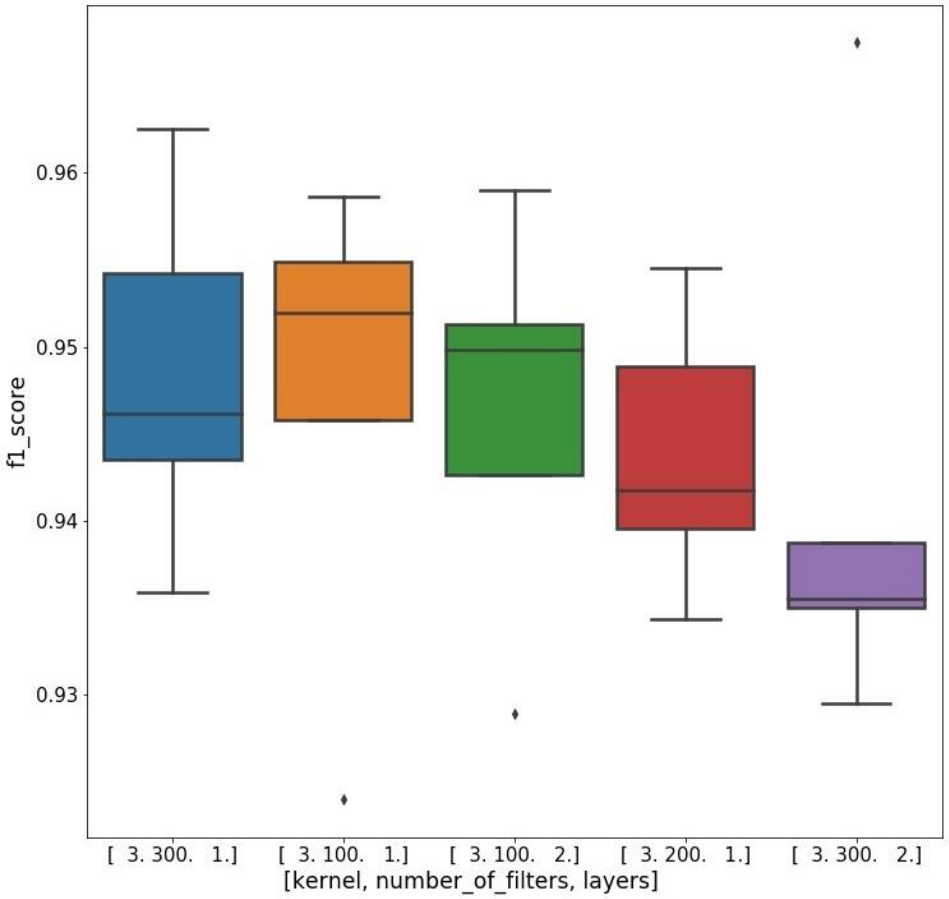

Рис. 3.31. Boxplot для ЗНМ (П. Порошенко)

Зведемо результати до таблиці.

Таблиця 3.1 Результи роботи алгоритмів

	В. Зеленський	П. Порошенко
Наївний Байєсівський класифікатор (Bag-of-Words)	94%	90%
Наївний Байєсівський класифікатор (TF-IDF)	94.2%	92%
SVM (Bag-of-Words)	94.5%	90%
SVM (TF-IDF)	93%	94%
Згорткова нейронна мережа	95.6%	95.5%

Отже, всі алгоритми досить точно класифікують дані при правильно підібраних параметрах.

Порівняємо найкращі моделі для кожного з кандидатів за допомогою коробкового графіку (boxplot). З рис. 3.32-3.33 очевидно, що найкращою моделлю є ЗНМ.

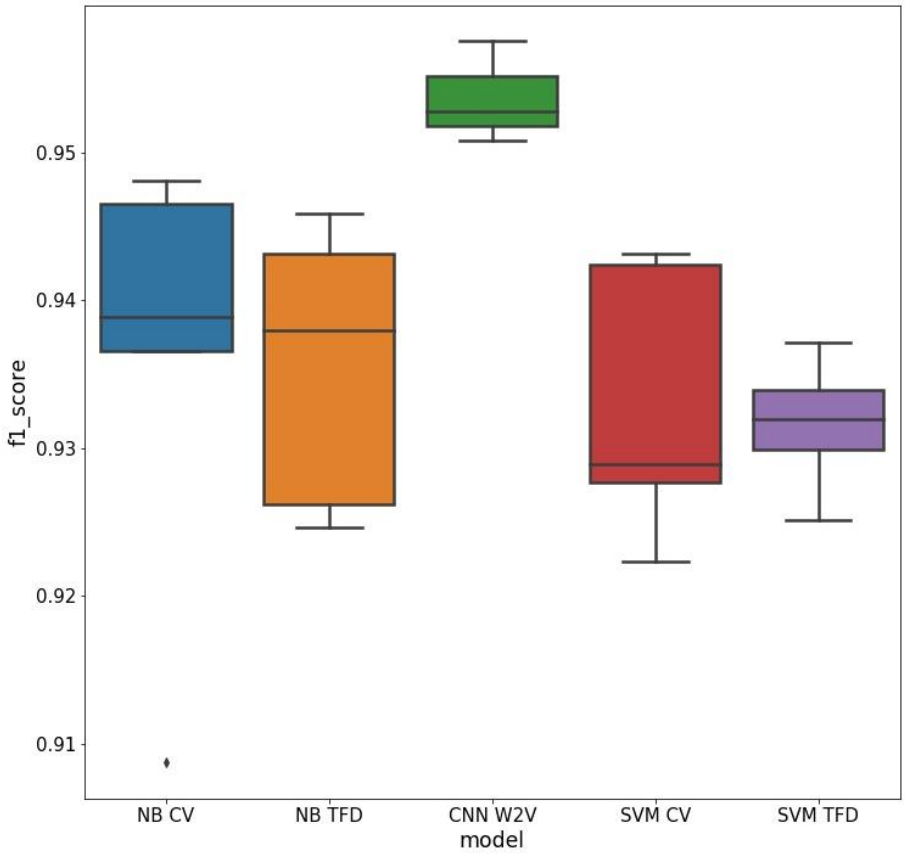

Рис. 3.32. Boxplot для найкращих моделей (В. Зеленський)

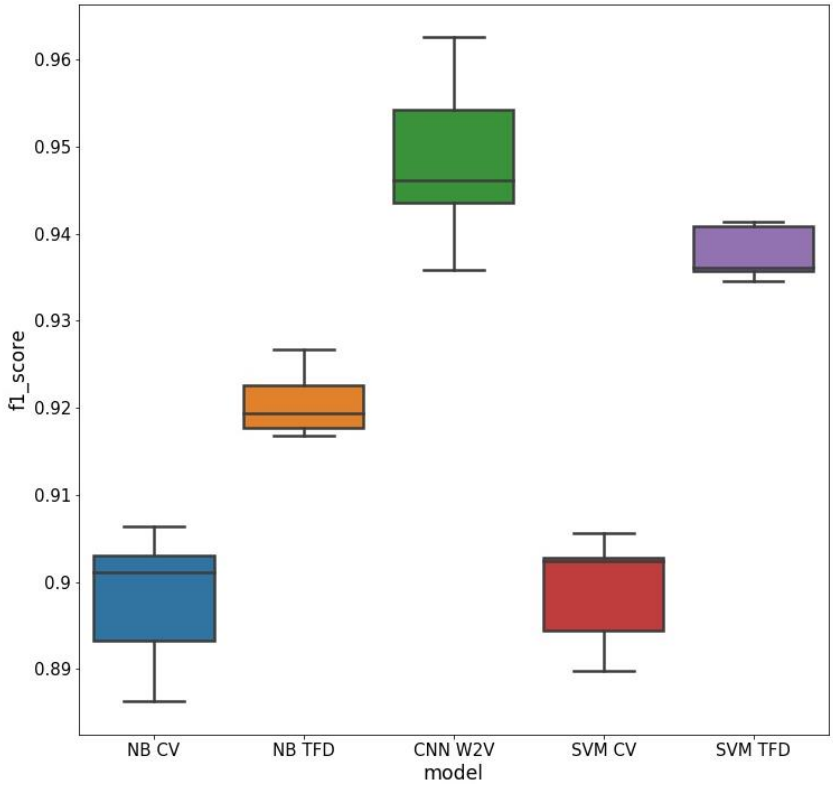

Рис. 3.33. Boxplot для найкращих моделей (П. Порошенко)

Всі моделі мають точність більше 90% (за метрикою F1). Найкращий результат для обох вибірок даних показала згорткова нейронна мережа з одним шаром згортки — точність 95.5%.

3.5. Аналіз зміни громадської думки

Визначивши найкращу модель, проаналізуємо з її допомогою зміну прихильності громадськості до кандидата за час передвиборчої кампанії. Для цього використаємо нерозмічені дані коментарів під публікаціями кандидатів в період з початку президентських перегонів до 2го туру виборів (03.01.2019 – 21.04.2019). Далі використаємо раніше навчену ЗНМ для класифікації коментарів. Результати класифікації (відсоток позитивного класу) зобразимо на графіку (рис. 3.34-3.35).

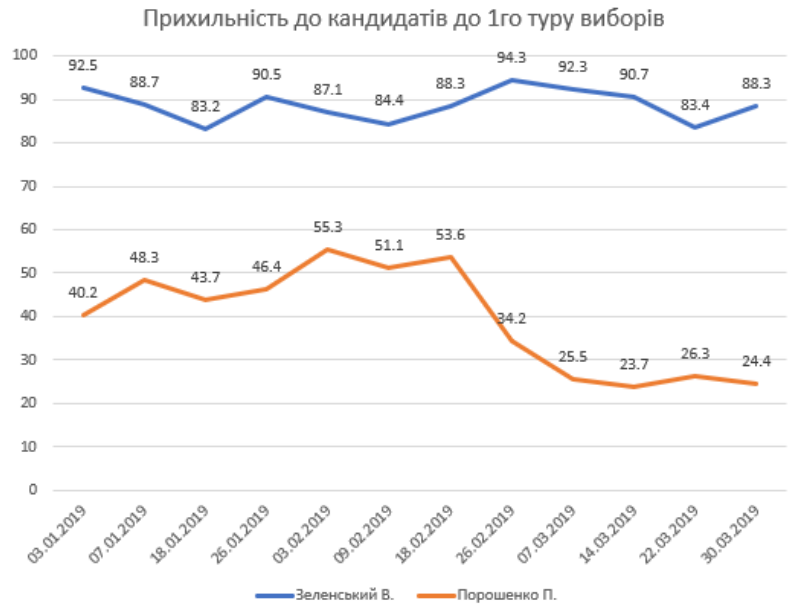

Рис. 3.34. Графік прихильності до кандидатів в період з 03.01.2019 по 30.03.2019

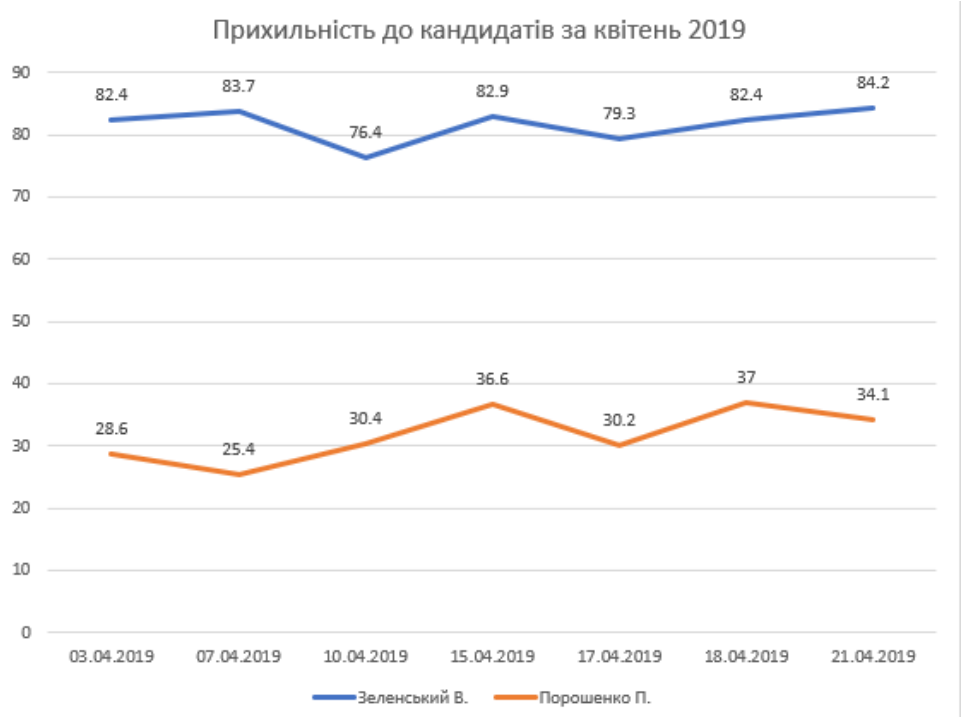

Рис. 3.35. Графік прихильності до кандидатів в період з 03.04.2019 по 21.04.2019

Графік дозволяє відслідковувати та аналізувати реакцію громадськості на події в автоматичному режимі. Наприклад на рис.3.32 на графіку В. Зеленського у точці за 18.01.2019 спостерігається різкий спад, який вірогідно пов’язано з оприлюдненням журналістського розслідування, в якому йдеться про те, що В. Зеленський через кіпрську фірму володіє кінокомпаніями в Росії. На тому ж рисунку на графіку П. Порошенка спостерігається різкий спад після 26.02.2019. Ця дата припадає на вихід журналістського розслідування про корупцію в «Укроборонпромі».

Висновки до розділу

У розділі проведено практичне дослідження, в якому оцінено зміну громадської думки перед першим та другим туром Президентських виборів у 2019 році.

Для навчання моделей було зібрано та промарковано близько 20 тис. записів, виконано аналіз та попередню обробку даних. Потім було навчено три моделі: НБК, SVM та згорткову нейронну мережу. Оптимальні параметри для моделей підбиралися шляхом кросвалідації. Точність моделей оцінювалась метрикою F1.

У результаті дослідження, всі моделі показали досить високу точність класифікації. Найбільш точним алгоритмом для даних обох кандидатів виявлась ЗМН з одним згортковим шаром (точність 95.5%).

Далі було проведено дослідження зміни громадської думки перед першим та другим туром Президентських виборів у 2019 році. Для цього було зібрано близько 70 тис. записів з публікацій кандидатів та класифіковано їх за допомогою раніше навченої ЗНМ.

РОЗДІЛ 4. РОЗРОБКА СТАРТАП ПРОЕКТУ

4.1 Вступ

Тенденція останніх років засвідчила, що зростаюча кількість малих інноваційних стартап-компаній, які працюють через Інтернет-платформи, формують майбутнє бізнесу. Тим не менш, статистика засвідчує, що дев'ять з десяти стартапів закінчуються невдачею. Вчені дослідили, що основною причиною невдач (близько 42% випадків відмови) є відсутність попиту на створений товар чи послугу [26].

Ефективним інструментом для запобігання провалу проекту на початкових етапах є ретельне дослідження ринку та складання бізнес-плану, який визначає бізнес-ідею, управлінський підхід та бізнес-стратегію.

Для розробки стартап проекту та виведення його на ринок необхідно провести детальне дослідження, яке передбачає виконання такх кроків.

1. Здійснити маркетинговий аналіз стартап-проекту, в рамках якого:

- розробити опис ідеї проекту, визначити основні напрямки використання товару чи послуги та сформулювати основні відмінності від товарів/послуг конкурентів;
- проаналізувати ринкові можливості для його реалізації;
- розробити стратегію виведення товару на ринок базуючись на аналізі ринкового середовища.

2. Організація стартап-проекту, яка включає такі кроки:

- скласти календарний план реалізації та запуску стартап-проекту;
- визначити плановий обсяг виробництва потенційного товару та на його основі розрахувати потребу у матеріальних ресурсах і персоналі;
- розрахувати витрати, необхідні для реалізації проекту, та витрати на запуск проекту.

3. Виконати фінансово-економічний аналіз та оцінити ризики стартап-проекту, в межах якого:

- визначити обсяг інвестиційних витрат;
- розрахувати основні фінансово-економічні показники проекту (собівартість, ціну продукту/послуги, податковий збір та чистий прибуток) та визначити показники інвестиційної привабливості проекту (рентабельність продажів, період окупності проекту);
- визначити основні ризики проекту та способи для їх запобігання.

4. Розробити заходи з комерціалізації проекту.

Цей етап націлений на пошук фінансування проекту та просування інвестиційної пропозиції. Для його досягнення необхідно:

- визначити цільову групу інвесторів та описати їх бізнес інтереси;
- скласти інвестиційну пропозицію: стислий опис проекту для ознайомлення інвестора із стартап-проектом;
- визначити основні канали та заходи для просування оферти інвесторам.

4.2. Опис ідеї проекту

Стартап проект полягає у створенні веб застосунку для автоматичного визначення тональності тексту. Така програма аналізує масиви даних, після чого дає користувачу звіт доповнений графіками.

В таблиці 4.1 представлено результати маркетингового дослідження стартап-проекту, а саме: сформульовано ідею та зміст проекту, напрямки застосування потенційного продукту та визначено переваги для користувача.

Таблиця 4.2.1. Опис ідеї стартап-проекту

Опис ідеї стартап проекту	Сфери застосування	Вигоди для користувача
Створення системи автоматичного аналізу тональності тексту	Аналіз текстових коментарів у соціальних мережах для визначення ставлення користувача до предмету свого висловлювання	Візуалізація результатів аналізу, цілісність системи, відсутність необхідності встановлювати додаткове ПЗ на пристрій

Далі було визначено конкурентів на ринку і зроблено порівняльний аналіз програмних продуктів конкурентів, виявлено їх переваги та недоліки. Також було представлено перелік переваг над існуючими програмними рішеннями.

Виконаний аналіз включає:

1. Складання переліку характеристик потенційного продукту.
2. Пошук і аналіз конкурентних продуктів, товарів-замінників чи товарів-аналогів, які вже знаходяться на ринку.
3. Порівняння властивостей та показників за такими критеріями: а) слабкі сторони; б) нейтральні сторони; в) сильні сторони.

Результат аналізу представлено у таблиці 4.2.2.

Таблиця 4.2.2. Визначення характеристик ідеї проекту

Техніко-економічні характеристики ідеї	Продукція конкурентів		Слабкі (W), нейтральні (N) та сильні (S) сторони		
Назва продукту	LexisNexis	Texterra	W	N	S
Операційна система та версії	Windows, MacOS, Linux	Windows		+	+
Системні вимоги	Мінімальні	Мінімальні			+
Необхідність встановлення додаткового ПЗ	наявність АПК	наявність АПК		+	
Мови, які аналізуються	Англійська	Англійська, російська	+		
Необхідний рівень знань для використання ПЗ	Немає	Наявність знань програмування	+		
Ціна	4500 грн	безкоштовний			+

Наразі на ринку відсутні повнофункціональні та кросплатформенні аналоги стартап-проекту.

Програмний продукт даного стартап-проекту представляє собою клієнт-серверний застосунок з дружнім та інтуїтивно зрозумілим інтерфейсом. Також немає необхідності встановлювати додаткове програмне забезпечення.

4.3. Опис технологічного аудиту проекту

Було здійснено перевірку технологій реалізації проекту. Аудит визначає чи потенційно можливо вдосконалити технології. Результат наведений у таблиці 4.3.1.

Таблиця 4.3.1. Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Створення системи автоматичної оцінки тональності тексту	VS Code, Jupyter, Python, React	+	+
	БД SQL Lite	+	+

Jupyter — середовище розробки, яке використовувалось для побудови моделей для аналізу тексту.

VS Code — середовище розробки, в якому було написано клієнтську та серверну частину застосунку.

Python — мова програмування, що використовувалась для побудови та навчання моделей аналізу тексту (pandas, numpy, sklearn) і написання серверної частини застосунку (flask).

React — JavaScript бібліотека для розробки інтерфейсу користувача.

Перелічені вище технології є відкритими і безкоштовними.

4.4. Аналіз ринкових можливостей запуску стартап-проекту

Аналіз ринкових можливостей дозволяє вибудувати стратегію виходу проекту на ринок з урахуванням ринкового середовища, потреб клієнтів і продуктів конкурентів та дає змогу виявити ринкові можливості і загрози, що можуть сприяти або перешкоджати успішному запуску продукту.

Перш за все проводиться аналіз потенційного ринку: наявність попиту, обсяг, рентабельність, динаміка розвитку ринку, обмеження (таблиця 4.4.1).

Таблиця 4.4.1. Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку	Характеристика
Кількість конкурентів	2
Загальна потреба в продукті	Гостра необхідність
Річні обсяги продажу	0.8млн $
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу	Немає
Специфічні вимоги до стандартизації та сертифікації	Немає
Середня норма рентабельності в галузі (або по ринку)	30%

Після аналізу ринку можна зробити висновок, що він є сприятливим для створення програмного продукту, оскільки динаміка ринку позитивна, а конкуренти майже відсутні, зважаючи, що основна маса ПЗ орієнтується на англомовні або російськомовні тексти.

Наступним кроком необхідно охарактеризувати основні групи потенційних користувачів продукту і скласти опис вимог кожної такої групи (таблиця 4.4.2).

Таблиця 4.4.2. Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія	Особливості поведінки споживачів	Вимоги споживачів до товару
Визначення настроїв, думок аудиторії стосовно певного об’єкту	Політтехнологи	Аналіз думок і настроїв виборців	Зручність інтерфейсу, візуалізація результатів аналізу
	Маркетологи	Аналіз думок користувачів товарів чи послуг, торгових марок для кращого розуміння аудиторії і коректнішої побудови маркетингових кампаній	Швидкість роботи; інтуїтивно-зрозумілий інтерфейс; кросплатформенність
	Соціологи	Виявлення настроїв населення, їх вподобань, ставлення до певного явища/об’єкту	Візуалізація результатів, можливість їх збереження на ПК, зручний і зрозумілий інтерфейс

Далі необхідно проаналізувати можливі загрози, що можуть виникнути на етапі виведення продукту на споживацький ринок і перешкодити успішному запуску проекту.

Результати представлені у таблиці 4.4.3.

Таблиця 4.4.3. Фактори загроз

Фактор	Зміст загрози	Можлива реакція компанії
Поява конкурентів	Поява аналогічного продукту на ринку, або поява схожого продукту з більш широким функціоналом або з нижчою ціною	Збільшення цінності та якості продукту для користувача шляхом створення нового функціоналу, оптимізація вартості, аналіз ринкових потреб
Зміни тенденцій ринку	Поява систем для виявлення конкретних емоцій у тексті (радість, нудьга, гнів тощо)	Розробка і впровадження моделі для виокремлення конкретних емоцій
Економічний спад	Кількісне зменшення клієнтної бази	Впровадження тестового періоду для ознайомлення користувача з продуктом, розробка системи знижок на ПЗ
Зниження репутації компанії	Проблеми з використанням ПЗ, якість наданих послуг	Виявлення причин зниження репутації та їх усунення: покращення системи тех-підтримки, покращення якості продукту, усунення багів

Також необхідно розглянути можливі фактори, що навпаки сприятимуть запуску проекту. Результати представлені у таблиці 4.4.4.

Таблиця 4.4.4. Фактори можливостей

Фактор	Зміст можливості	Можлива реакція компанії
Відсутність сильної конкуренції	На даний момент на ринку відсутні системи для аналізу тональності україномовних текстів	Розширення можливостей продукту, вихід на іноземні ринки, інтеграція з іншими програмними продуктами
Залежність від ринкових потреб	На сьогодні важливою частиною життя кожного є соціальні мережі, у яких користувачі тим чи іншим чином висловлюють свої настрої та вподобання, що є відмінною ареною впливу для досягнення певною мети	Розміщення реклами на сторінках найпопулярніших соцмереж
Створення позитивного іміджу компанії	Надання послуг на найвищому рівні, забезпечення задоволення клієнтів	Створення якісної рекламної кампанії, техпідтримка існуючих клієнтів

Далі необхідно охарактеризувати конкурентне середовище, а саме визначити тип та рівень конкуренції. Результати аналізу наведено у таблиці 4.4.5.

Таблиця 4.4.5. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії)
Зазначення типу конкуренції (досконала, недосконала: монополія, олігополія)	Олігополія	Позитивна репутація компанії та висока якість продукту забезпечить перевагу перед конкурентами
За рівнем конкурентної боротьби (локальний, глобальний)	Глобальний	Розширення функціоналу системи для виходу на нові ринки і охоплення більшої частки ринку
За галузевою ознакою (міжгалузева, внутрішньогалузева)	Внутрішньогалузева	—
Конкуренція за видами товарів (товарно-родова, товарно-видова, між бажаннями)	Товарно-видова	—
За характером конкурентних переваг (цінова, нецінова)	Нецінова	Створити найбільшу цінність продукту серед конкурентів
За інтенсивністю (марочна, не марочна)	Немарочна	Забезпечення оптимальної ціни, високої якості

Далі необхідно виконати детальний аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера, яка використовується для розуміння структури галузі, аналізу її привабливості з точки зору отримання прибутку, оцінки конкуренції і розробки стратегії бізнесу. Результати аналізу зведено в таблицю 4.4.6.

Таблиця 4.4.6. Аналіз конкуренції в галузі за моделлю Майкла Портера

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Перелік прямих конкурентів	Бар'єри входження в ринок	Фактори сили постачальників	Фактори сили споживачів	Загрози з боку замінників
Висновки	На ринку є два конкуренти, що мають схожий продукт	Для входження на ринок необхідно мати капітал для розробки і запуску системи	Постачальники відсутні	Споживачами даного продукту є бізнес клієнти; можуть беззатратно перейти до конкурентів	Товари-замінники відсутні

Результати аналізу конкурентного середовища підтверджують, що на ринку сприятлива ситуація для створення і запуску даного стартап-проекту.

Ґрунтуючись на проведеному аналізі конкуренції (таблиця 4.4.6), а також враховуючи характеристики ідеї стартап-проекту (таблиця 4.2.2), характеристики потенційних клієнтів і їх вимоги до продукту (таблиця 4.4.2) та

фактори ринкового середовища (таблиці 4.4.3 та 4.4.4) було сформульовано та обґрунтовано перелік факторів конкурентоспроможності. Аналіз оформлено в таблицю 4.4.7.

Таблиця 4.4.7. Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування
Низька конкуренція	В нашій країні на час розробки стартапу не було виявлено конкурентів
Доступність програмного продукту	Розроблений продукт є загальнодоступним і кросплатформенним. Для доступу необхідне підключення до мережі Інтернет
Зручність використання	Дружній і зрозумілий інтерфейс, від користувачів не потребується наявність спеціальних навичок для використання сервісу

Після проведення аналізу можна виділити сильні та слабкі (які потребують вдосконалення) сторони продукту (таблиця 4.4.8).

Таблиця 4.4.8. Порівняльний аналіз сильних та слабких сторін проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
		–3	–2	–1	0	+1	+2	+3
Невелика кількість конкурентів	18				+			
Доступність програмного продукту	17			+				
Зручність використання	17		+					

Завершальним етапом аналізу ринкових можливостей для запуску проекту є складання SWOT-аналізу. Він дозволяє оцінити можливості та загрози бізнесу, а також сильні і слабкі сторони продукту. Результати наведені у таблиці 4.4.9.

Таблиця 4.4.9. SWOT-аналіз проекту

Сильні сторони (S): відсутність конкурентів; дружній інтерфейс; не вимагає спеціальних навичок/знань для використання	Слабкі сторони (W): залежність від предметної області
Можливості (O): розширення списку мов для аналізу додавання нових предметних областей інтеграція з іншими програмними системами	Загрози (T): поява конкурентів; погіршення економічної ситуації

На основі SWOT-аналізу було спроектовано альтернативну ринкову поведінку для інтеграції стартап-проекту на ринок та приблизний час реалізації системного комплексу, з урахуванням потенційних проектів, що можуть бути виведені на ринок (таблиця 4.4.10).

Таблиця 4.4.10. Альтернативи ринкового впровадження стартап-проекту

Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Вихід на міжнародний ринок (розробка дизайну, програмного комплексу, маркетингові зусилля для просування продукту)	85%	10 місяців
Інтеграція з існуючими програмними системами	75%	6 місяців

Отже, в результаті детального аналізу ринкового та конкурентного середовища, факторів загроз та можливостей, сильних та слабких сторін продукту можна зробити висновок, що на ринку склалися сприятливі умови для впровадження товару і, що даний товар відповідає вимогам користувачів.

4.5. Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії перш за все передбачає визначення стратегії охоплення ринку. Для цього було охарактеризовано цільові групи потенційних споживачів (таблиця 4.5.1).

Таблиця 4.5.1. Вибір цільових груп потенційних споживачів

Опис цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в сегменті	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Політтехнологи	Висока	Високий попит	Відсутня	Низька
Соціологи	Середня	Середній попит	Відсутня	Низька
Маркетологи	Висока	Високий попит	Середня	Середня

Проаналізувавши три потенційні групи споживачів, можна зробити висновок, що вони не мають суттєвих відмінностей, тому в даному продукті буде реалізовано стандартний пакет функціоналу для кожної з них. Стратегією охоплення ринку було обрано недиференційований (масовий) маркетинг — компанія концентрує свої зусилля одразу на всіх сегментах споживачів.

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 4.5.2, 4.5.3, 4.5.4).

Таблиця 4.5.2. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Розробка програмного продукту, просування	Концентрований маркетинг	Доступність, легкість у використанні	Масовий маркетинг

Таблиця 4.5.3. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопроходцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні	Шукати нових	Ні	Стратегія наслідування

Таблиця 4.5.4. Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Легкість у використанні, висока точність аналізу	Позиціювання як продукту, що використовує передові технології для аналізу	Висока точність, візуалізація результатів	Високотехнологічність, зручність, легкість у використанні

Висновки до розділу

У результаті виконання даного розділу магістерської дисертації було з’ясовано, що існує реальна можливість ринкової комерціалізації розробленого продукту. Також слід зазначити, що даний продукт буде рентабельним, оскільки на україномовному ринку відсутні аналоги і конкуренція.

Зважаючи на потенційних групи споживачів, варто зазначити, що вони мають багато спільного, тому стратегією охоплення ринку було обрано недиференційований (масовий) маркетинг. Сервіс матиме стандартний набір функціоналу, що задовольнятиме всі потреби клієнтів.

ВИСНОВКИ

У магістерській дисертації було проведено огляд основних підходів до вирішення задачі аналізу тональності тексту. Також було зроблено огляд методів машинного навчання, що використовуються для сентимент аналізу тексту та обрано три з них для проведення практичного дослідження.

Дослідження полягало у визначенні тональності тексту коментарів під публікаціями кандидатів у Президенти України (В.Зеленський та П.Порошенко) в період передвиборчих перегонів 2019 року (перед першим та другим турами).

Для визначення тональності тексту було використано три алгоритма: наївний Байєсівський класифікатор, метод опорних векторів та згорткова нейронна мережа.

Для переведення тексту у вектор було використано три алгоритма переведення тексту в векторний вигляд – Bag-of-Words, TF-IDF та Word2Vec. Для кожного з кандидатів було побудовано окремі моделі і порівняно якість класифікації (за метрикою F1). Загалом всі моделі показали досить високу точність при правильно підібраних параметрах. Найкращою моделлю, для обох вибірок даних виявилась згорткова нейронна мережа з одним шаром згортки, 300 фільтрами, коефіцієнтом дропаута 0.4.

Найкращу модель було застосовано для аналізу зміни громадської думки в період з 03.01.2019 по 21.04.2019.

Як перспективу до побальших досліджень ми вбвчаємо проведення сентимент аналізу на рівні об'єкту-аспекту.

СПИСОК ЛІТЕРАТУРИ

1. Nasukawa T. Sentiment analysis: Capturing favorability using natural language processing /Nasukawa T., Yi J. // Proc. of the 2nd Int. Conf. on Knowledge capture (KCAP), 2003. P. 7077.

2. Dave K. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews /Dave K., Lawrence St., Pennock D. //Proc. of the Int. Conf. on World Wide Web (WWW), 2003. P. 519528.

3. Барсегян А. Технологии анализа данных: Data Mining, Text Mining, Visual Mining, OLAP / Барсегян А. — 2 изд. — БХВ-Петербург, 2008. — 384 р.

4. Vimala Balakrishnan. Stemming and Lemmatization: A Comparison of Retrieval Performances /Vimala Balakrishnan, 2014. — 204 p.

5. Нога Р. Аналітичний огляд методів та засобів опрацювання текстової інформації /Р.Нога, Н.Б.Шаховська //Вісник національного університету «Львівська політехніка». — 2011. — С. 323—332.

6. Сахарный Л.В., Штерн А.С. Набор ключевых слов как тип текста / Сахарный Л.В., Штерн А.С. //Лексические аспекты в системе профессионально-ориентированного обучения иноязычной речевой деятельности. — Пермь: Пермский политехнический ун-т, 1988. — С. 34—51.

7. Яцко В.А. Лексикографические ресурсы для автоматического анализа текста [Текст] /В.А.Яцко //Вестник Иркутского государственного лингвистического университета. — 2013. — № 2. — С. 19—24.

8. Гамма Эрих. Приемы объектно-ориентированного проектирования /Эрих Гамма. — Издательский дом «Питер», 2013. — 368 с.

9. Основы вычислительного интеллекта, М. З. Згуровский, Ю.П. Зайченко. – Изд. «Наукова думка». Киев. – 2013. – 406 с.

10. Aggarwal Charu C, Zhai Cheng Xiang. Mining Text Data. – Springer New York Dordrecht Heidelberg London: Springer Science+Business Media, 2012.

11. Kaufmann JM. JMaxAlign: A Maximum Entropy Parallel Sentence Alignment Tool. – In: Proceedings of COLING'12: Demonstration Papers, Mumbai; 2012. – 88 p.

12. J.R. Quinlan. Induction of decision trees Machine Learn. – Kluwer Academic Publishers, Boston, 1986. – pp. 81-106.

13. David D. Lewis, Marc Ringuette. A comparison of two learning algorithms for text categorization. – SDAIR, 1994.

14. Soumen Chakrabarti, Shourya Roy, Mahesh V. Soundalgekar. Fast and accurate text classification via multiple linear discriminant projections. – VLDB J, 2, 2003. – pp. 172-185.

15. Alexis Conneau, Holger Schwenk, Loïc Barrault, Yann Lecun. Very Deep Convolutional Networks for Text Classification. – Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 2017.

16. X Fulin, D Yihao, T Xiaosheng. The Architecture of Word2vec and Its Applications. – Journal of Nanjing, 2015.

17. L Du, Y Wang, G Song, Z Lu, J Wang. Dynamic Network Embedding: An Extended Approach for Skip-gram based Network Embedding. – IJCAI, 2018.

18. Bing Liu. Sentiment Analysis and Opinion Mining / Bing Liu. — Morgan & Claypool Publishers, May 2012.

19. Vincent Ng. Weakly Supervised Natural Language Learning Without Redundant Views / Vincent Ng, Claire Cardie. — 2003.

20. Toby Segaran. Programming collective intelligence /Toby Segaran. — O'Reilly, 2007.

21. E. Boiy. Automatic Sentiment Analysis in On-line Text / E. Boiy, P. Hens, K. Deschacht. — Vienna, 2007.

22. Bo Pang, Lillian Lee. Opinion Mining and Sentiment Analysis /Bo Pang, Lillian Lee. — 2008.

23. Bo Pang. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts / Bo Pang, Lillian Lee. — 2004.

24. Wiebe, Bruce. Development and use of a gold-standard data set for subjectivity classifications /Wiebe, Bruce and O'Hara. — 1999.

25. Jindal, Liu. Mining comparative sentences and relations /Jindal, Liu. — 2006

26. CB Insights. The Top 20 Reasons Startups Fail, 2019. — Джерело доступу: https://www.cbinsights.com/research/startup-failure-reasons-top/

27. Tomas Mikolov. Efficient estimation of word representations in vector space /Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. — arXiv preprint arXiv:1301.3781. ICLR Workshop, 2013. — P. 1—12.

28. Y.Kim. Convolutional neural networks for sentence classification /Y.Kim //Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, Association for Computational Linguistics, October 2014. P. 1746—1751.

29. P.D.Turney. Thumbs up or thumbs down: semantic orientation applied to unsupervised classification of reviews /P.D.Turney //Proceedings of the 40th annual meeting on association for computational linguistics. — 2002. — P. 417–420.

30. T.Wilson. Recognizing contextual polarity in phrase-level sentiment analysis /T.Wilson, J.Wiebe, and P.Hoffmann //Proceedings of the conference on human language technology and empirical methods in natural language processing. — 2005. — P. 347—354.

31. Liu, Bing. Sentiment analysis and subjectivity /Liu, Bing // Handbook of natural language processing. — Boca Raton: CRC Press, 2010. 2nd ed. — P. 627—666.

32. The General Inquirer: A Computer Approach to Content Analysis. /Stone Philip J., Dexter C. Dunphy, Marshall S. Smith, and Daniel M // Ogilvie. MIT Press, Cambridge, MA, 1966. 35 p.

33. G.Katz. Context-based sentiment analysis / G. Katz, N. Ofek, and B. Shapira // Knowledge-Based Systems. ConSent. — 2015. Vol. 84, No. 1. P. 162—178.

34. Rodrigo Moraes. Document-level sentiment classification: an empirical comparison between SVM and ANN / Rodrigo Moraes, João Valiati, Wilson P. //Expert Syst. Appl, 40. — 2013. — P. 621—633.

35. R. Feldman. Techniques and applications for sentiment analysis /R. Feldman //Communications of the ACM, 2013.

36. B. Liu. Sentiment analysis and subjectivity /B.Liu //Handbook of natural language processing, 2010.

37. AL Maas. Learning word vectors for sentiment analysis /AL Maas, RE Daly, PT Pham, D Huang //Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. vol. 1, June 2011. — pp. 142—150.

38. M Taboada. Lexicon-based methods for sentiment analysis /M Taboada, J Brooke, M Tofiloski, K Voll //Computational Linguistics. MIT Press, 2011. — pp. 267—307.

39. N Sebe. Emotion recognition using a cauchy naive bayes classifier /N Sebe, MS Lew, I Cohen, A Garg //IEEE, Quebec, 2002.